svy estimation — Estimation commands for survey data

Description

Survey data analysis in Stata is essentially the same as standard data analysis. The standard syntax applies; you just need to also remember the following:

- Use `svyset` to identify the survey design characteristics.
- Prefix the estimation commands with `svy:`.

For example,

```
. use https://www.stata-press.com/data/r16/nhanes2f
. svyset psuid [pweight=finalwgt], strata(stratid)
. svy: regress zinc age c.age#c.age weight female black orace rural
```

See [SVY] `svyset` and [SVY] `svy`.

The following estimation commands support the `svy` prefix:

Descriptive statistics

- `mean` ([R] `mean` — Estimate means)
- `proportion` ([R] `proportion` — Estimate proportions)
- `ratio` ([R] `ratio` — Estimate ratios)
- `total` ([R] `total` — Estimate totals)

Linear regression models

- `churdle` ([R] `churdle` — Cragg hurdle regression)
- `cnsreg` ([R] `cnsreg` — Constrained linear regression)
- `eintreg` ([ERM] `eintreg` — Extended interval regression)
- `eregress` ([ERM] `eregress` — Extended linear regression)
- `etregress` ([TE] `etregress` — Linear regression with endogenous treatment effects)
- `glm` ([R] `glm` — Generalized linear models)
- `hetregress` ([R] `hetregress` — Heteroskedastic linear regression)
- `intreg` ([R] `intreg` — Interval regression)
- `nl` ([R] `nl` — Nonlinear least-squares estimation)
- `regress` ([R] `regress` — Linear regression)
- `tobit` ([R] `tobit` — Tobit regression)
- `truncreg` ([R] `truncreg` — Truncated regression)

Structural equation models

- `sem` ([SEM] `sem` — Structural equation model estimation command)
- `gsem` ([SEM] `gsem` — Generalized structural equation model estimation command)
Survival-data regression models

- `stcox`  
  — Cox proportional hazards model
- `stintreg`  
  — Parametric models for interval-censored survival-time data
- `streg`  
  — Parametric survival models

Binary-response regression models

- `biprobit`  
  — Bivariate probit regression
- `cloglog`  
  — Complementary log-log regression
- `eprobit`  
  — Extended probit regression
- `hetprobit`  
  — Heteroskedastic probit model
- `logistic`  
  — Logistic regression, reporting odds ratios
- `logit`  
  — Logistic regression, reporting coefficients
- `probit`  
  — Probit regression
- `scobit`  
  — Skewed logistic regression

Discrete-response regression models

- `clogit`  
  — Conditional (fixed-effects) logistic regression
- `cmmixlogit`  
  — Mixed logit choice model
- `cmxtnmixlogit`  
  — Panel-data mixed logit choice model
- `eoprobit`  
  — Extended ordered probit regression
- `hetoprobit`  
  — Heteroskedastic ordered probit regression
- `mlogit`  
  — Multinomial (polytomous) logistic regression
- `mprobit`  
  — Multinomial probit regression
- `ologit`  
  — Ordered logistic regression
- `oprobit`  
  — Ordered probit regression
- `slogit`  
  — Stereotype logistic regression
- `zioprobit`  
  — Zero-inflated ordered probit regression

Fractional-response regression models

- `betareg`  
  — Beta regression
- `fracreg`  
  — Fractional response regression

Poisson regression models

- `cpoisson`  
  — Censored Poisson regression
- `etpoisson`  
  — Poisson regression with endogenous treatment effects
- `gnbreg`  
  — Generalized negative binomial regression in [R] `nbreg`
- `nbreg`  
  — Negative binomial regression
- `poisson`  
  — Poisson regression
- `tnbreg`  
  — Truncated negative binomial regression
- `tpoisson`  
  — Truncated Poisson regression
- `zinb`  
  — Zero-inflated negative binomial regression
- `zip`  
  — Zero-inflated Poisson regression

Instrumental-variables regression models

- `ivprobit`  
  — Probit model with continuous endogenous covariates
- `ivregress`  
  — Single-equation instrumental-variables regression
- `ivtobit`  
  — Tobit model with continuous endogenous covariates
Regression models with selection

heckman  [R] heckman — Heckman selection model
heckoprobit  [R] heckoprobit — Ordered probit model with sample selection
heckpoisson  [R] heckpoisson — Poisson regression with sample selection
heckprobit  [R] heckprobit — Probit model with sample selection

Multilevel mixed-effects models

celoglog  [ME] celoglog — Multilevel mixed-effects complementary log-log regression
celm  [ME] celm — Multilevel mixed-effects generalized linear model
cemaintr  [ME] cemaintr — Multilevel mixed-effects interval regression
celogit  [ME] celogit — Multilevel mixed-effects logistic regression
cemnegbin  [ME] cemnegbin — Multilevel mixed-effects negative binomial regression
cemlogit  [ME] cemlogit — Multilevel mixed-effects ordered logistic regression
cemprobit  [ME] cemprobit — Multilevel mixed-effects ordered probit regression
cempoisson  [ME] cempoisson — Multilevel mixed-effects Poisson regression
cemprobit  [ME] cemprobit — Multilevel mixed-effects probit regression
cemreg  [ME] cemreg — Multilevel mixed-effects parametric survival models
cemtobit  [ME] cemtobit — Multilevel mixed-effects tobit regression

Finite mixture models

fmm: betareg  [FMM] fmm: betareg — Finite mixtures of beta regression models
fmm: cloglog  [FMM] fmm: cloglog — Finite mixtures of complementary log-log regression models
fmm: glm  [FMM] fmm: glm — Finite mixtures of generalized linear regression models
fmm: intreg  [FMM] fmm: intreg — Finite mixtures of interval regression models
fmm: ivregress  [FMM] fmm: ivregress — Finite mixtures of linear regression models with endogenous covariates
fmm: logit  [FMM] fmm: logit — Finite mixtures of logistic regression models
fmm: mlogit  [FMM] fmm: mlogit — Finite mixtures of multinomial (polytomous) logistic regression models
fmm: nbreg  [FMM] fmm: nbreg — Finite mixtures of negative binomial regression models
fmm: ologit  [FMM] fmm: ologit — Finite mixtures of ordered logistic regression models
fmm: oprobit  [FMM] fmm: oprobit — Finite mixtures of ordered probit regression models
fmm: pointmass  [FMM] fmm: pointmass — Finite mixtures models with a density mass at a single point
fmm: poisson  [FMM] fmm: poisson — Finite mixtures of Poisson regression models
fmm: probit  [FMM] fmm: probit — Finite mixtures of probit regression models
fmm: regress  [FMM] fmm: regress — Finite mixtures of linear regression models
fmm: streg  [FMM] fmm: streg — Finite mixtures of parametric survival models
fmm: tobit  [FMM] fmm: tobit — Finite mixtures of tobit regression models
fmm: tpoisson  [FMM] fmm: tpoisson — Finite mixtures of truncated Poisson regression models
fmm: truncreg  [FMM] fmm: truncreg — Finite mixtures of truncated linear regression models

Item response theory

irt 1pl  [IRT] irt 1pl — One-parameter logistic model
irt 2pl  [IRT] irt 2pl — Two-parameter logistic model
Menu

Statistics > Survey data analysis > …

Dialog boxes for all statistical estimators that support `svy` can be found on the above menu path. In addition, you can access survey data estimation from standard dialog boxes on the SE/Robust or SE/Cluster tab.

Remarks and examples

Remarks are presented under the following headings:

* Overview of survey analysis in Stata
* Descriptive statistics
* Regression models
* Health surveys

Overview of survey analysis in Stata

Many Stata commands estimate the parameters of a process or population by using sample data. For example, `mean` estimates means, `ratio` estimates ratios, `regress` fits linear regression models, `poisson` fits Poisson regression models, and `logistic` fits logistic regression models. Some of these estimation commands support the `svy` prefix, that is, they may be prefixed by `svy:` to produce results appropriate for complex survey data. Whereas `poisson` is used with standard, nonsurvey data, `svy: poisson` is used with survey data. In what follows, we refer to any estimation command not prefixed by `svy:` as the standard command. A standard command prefixed by `svy:` is referred to as a `svy` command.

Most standard commands (and all standard commands supported by `svy`) allow `pweights` and the `vce(cluster clustvar)` option, where `clustvar` corresponds to the PSU variable that you `svyset`. If your survey data exhibit only sampling weights or first-stage clusters (or both), you can get by with using the standard command with `pweights`, `vce(cluster clustvar)`, or both. Your parameter estimates will always be identical to those you would have obtained from the `svy` command, and the standard command uses the same robust (linearization) variance estimator as the `svy` command with a similarly `svyset` design.

Most standard commands are also fit using maximum likelihood. When used with independently distributed, nonweighted data, the likelihood to be maximized reflects the joint probability distribution of the data given the chosen model. With complex survey data, however, this interpretation of the likelihood is no longer valid, because survey data are weighted, not independently distributed, or both. Yet for survey data, (valid) parameter estimates for a given model can be obtained using the associated likelihood function with appropriate weighting. Because the probabilistic interpretation no longer holds, the likelihood here is instead called a pseudolikelihood, but likelihood-ratio tests are no longer valid. See Skinner (1989, sec. 3.4.4) for a discussion of maximum pseudolikelihood estimators.
Here we highlight the other features of _svy_ commands:

- _svy_ commands handle stratified sampling, but none of the standard commands do. Because stratification usually makes standard errors smaller, ignoring stratification is usually conservative. So not using _svy_ with stratified sample data is not a terrible thing to do. However, to get the smallest possible “honest” standard-error estimates for stratified sampling, use _svy_.

- _svy_ commands use \( t \) statistics with \( n - L \) degrees of freedom to test the significance of coefficients, where \( n \) is the total number of sampled PSUs (clusters) and \( L \) is the number of strata in the first stage. Some of the standard commands use \( t \) statistics, but most use \( z \) statistics. If the standard command uses \( z \) statistics for its standard variance estimator, then it also uses \( z \) statistics with the robust (linearization) variance estimator. Strictly speaking, \( t \) statistics are appropriate with the robust (linearization) variance estimator; see [P] _robust_ for the theoretical rationale. But, using \( z \) rather than \( t \) statistics yields a nontrivial difference only when there is a small number of clusters (\(< 50\)). If a regression model command uses \( t \) statistics and the vce(cluster clustvar) option is specified, then the degrees of freedom used is the same as that of the _svy_ command (in the absence of stratification).

- _svy_ commands produce an adjusted Wald test for the model test, and _test_ can be used to produce adjusted Wald tests for other hypotheses after _svy_ commands. Only unadjusted Wald tests are available if the _svy_ prefix is not used. The adjustment can be important when the degrees of freedom, \( n - L \), is small relative to the dimension of the test. (If the dimension is one, then the adjusted and unadjusted Wald tests are identical.) This fact along with the point made in the second bullet make using the _svy_ command important if the number of sampled PSUs (clusters) is small (\(< 50\)).

- _svy: regress_ differs slightly from _regress_ and _svy: ivregress_ differs slightly from _ivregress_ in that they use different multipliers for the variance estimator. _regress_ and _ivregress_ (when the _small_ option is specified) use a multiplier of \( \{(N - 1)/(N - k)\}\{n/(n - 1)\} \), where \( N \) is the number of observations, \( n \) is the number of clusters (PSUs), and \( k \) is the number of regressors including the constant. _svy: regress_ and _svy: ivregress_ use \( n/(n - 1) \) instead. Thus they produce slightly different standard errors. The \( (N - 1)/(N - k) \) is ad hoc and has no rigorous theoretical justification; hence, the purist _svy_ commands do not use it. The _svy_ commands tacitly assume that \( N \gg k \). If \( (N - 1)/(N - k) \) is not close to 1, you may be well advised to use _regress_ or _ivregress_ so that some punishment is inflicted on your variance estimates. Maximum likelihood estimators in Stata (for example, _logit_) do no such adjustment but rely on the sensibilities of the analyst to ensure that \( N \) is reasonably larger than \( k \). Thus the maximum pseudolikelihood estimators (for example, _svy: logit_) produce the same standard errors as the corresponding maximum likelihood commands (for example, _logit_), but \( p \)-values are slightly different because of the point made in the second bullet.

- _svy_ commands can produce proper estimates for subpopulations by using the _subpop()_ option. Using an _if_ restriction with _svy_ or standard commands can yield incorrect standard-error estimates for subpopulations. Often an _if_ restriction will yield the same standard error as _subpop();_ most other times, the two standard errors will be slightly different; but sometimes—usually for thinly sampled subpopulations—the standard errors can be appreciably different. Hence, the _svy_ command with the _subpop()_ option should be used to obtain estimates for thinly sampled subpopulations. See [SVY] _Subpopulation estimation_ for more information.

- _svy_ commands handle zero sampling weights properly. Standard commands ignore any observation with a weight of zero. Usually, this will yield the same standard errors, but sometimes they will differ. Sampling weights of zero can arise from various postsampling adjustment procedures. If the sum of weights for one or more PSUs is zero, _svy_ and standard commands will produce different standard errors, but usually this difference is very small.
You can \texttt{svyset iweights} and let these weights be negative. Negative sampling weights can arise from various postsampling adjustment procedures. If you want to use negative sampling weights, then you must \texttt{svyset iweights} instead of \texttt{pweights}; no standard command will allow negative sampling weights.

The \texttt{svy} commands compute finite population corrections (FPCs).

After a \texttt{svy} command, \texttt{estat effects} will compute the design effects \texttt{DEFF} and \texttt{DEFT} and the misspecification effects \texttt{MEFF} and \texttt{MEFT}.

\texttt{svy} commands can perform variance estimation that accounts for multiple stages of clustered sampling.

\texttt{svy} commands can perform variance estimation that accounts for poststratification adjustments to the sampling weights.

Some standard options are not allowed with the \texttt{svy} prefix. For example, \texttt{vce()} and weights cannot be specified when using the \texttt{svy} prefix because \texttt{svy} is already using the variance estimation and sampling weights identified by \texttt{svyset}. Some options are not allowed with survey data because they would be statistically invalid, such as \texttt{noskip} for producing optional likelihood-ratio tests. Other options are not allowed because they change how estimation results are reported (for example, \texttt{nodisplay}, \texttt{first}, \texttt{plus}) or are not compatible with \texttt{svy}'s variance estimation methods (for example, \texttt{irls}, \texttt{mse1}, \texttt{hc2}, \texttt{hc3}).

Estimation results are presented in the standard way, except that \texttt{svy} has its own table header: In addition to the sample size, model test, and $R^2$ (if present in the output from the standard command), \texttt{svy} will also report the following information in the header:

a. number of strata and PSUs
b. number of poststrata, if specified to \texttt{svyset}
c. population size estimate
d. subpopulation sizes, if the \texttt{subpop()} option was specified
e. design degrees of freedom

**Descriptive statistics**

Use \texttt{svy: mean}, \texttt{svy: ratio}, \texttt{svy: proportion}, and \texttt{svy: total} to estimate finite population and subpopulation means, ratios, proportions, and totals, respectively. You can also estimate standardized means, ratios, and proportions for survey data; see \texttt{[SVY]} \textbf{Direct standardization}. Estimates for multiple subpopulations can be obtained using the \texttt{over()} option; see \texttt{[SVY]} \textbf{Subpopulation estimation}.

**Example 1**

Suppose that we need to estimate the average birthweight for the population represented by the National Maternal and Infant Health Survey (NMIHS) \textit{(Gonzalez, Krauss, and Scott 1992)}.

First, we gather the survey design information:

- Primary sampling units are mothers; that is, PSUs are individual observations—there is no separate PSU variable.
- The \texttt{finalwgt} variable contains the sampling weights.
- The \texttt{stratan} variable identifies strata.
- There is no variable for the finite population correction.
Then we use `svyset` to identify the variables for sampling weights and stratification.

```
. use https://www.stata-press.com/data/r16/nmihs
. svyset [pweight=finwgt], strata(stratan)
   
   pweight: finwgt
   VCE: linearized
   Single unit: missing
   Strata 1: stratan
   SU 1: <observations>
   FPC 1: <zero>
```

Now we can use `svy: mean` to estimate the average birthweight for our population.

```
. svy: mean birthwgt
   
   (running mean on estimation sample)
```

### Regression models

As exhibited in the table at the beginning of this manual entry, many of Stata’s regression model commands support the `svy` prefix. If you know how to use one of these commands with standard data, then you can also use the corresponding `svy` command with your survey data.

> **Example 2**

Let’s model the incidence of high blood pressure with a dataset from the Second National Health and Nutrition Examination Survey (NHANES II) (McDowell et al. 1981). The survey design characteristics are already `svyset`, so we will just replay them.

```
. use https://www.stata-press.com/data/r16/nhanes2d
. svyset
   
   pweight: finalwgt
   VCE: linearized
   Single unit: missing
   Strata 1: strata
   SU 1: psu
   FPC 1: <zero>
```

Now we can use `svy: logistic` to model the incidence of high blood pressure as a function of height, weight, age, and sex (using the `female` indicator variable).
The odds ratio for the `female` predictor is 0.63 (rounded to two decimal places) and is significantly less than 1. This finding implies that females have a lower incidence of high blood pressure than do males.

Here we use the `subpop()` option to model the incidence of high blood pressure in the subpopulation identified by the `female` variable.

```
.svy, subpop(female): logistic highbp height weight age
(running `logistic` on estimation sample)
```

Survey: Logistic regression
Number of strata = 31
Number of PSUs = 62
Number of obs = 10,351
Population size = 117,157,513
Design df = 31
F( 3, 29) = 227.53
Prob > F = 0.0000

|           | Odds Ratio | Std. Err. | t     | P>|t|  | [95% Conf. Interval] |
|-----------|------------|-----------|-------|------|----------------------|
| height    | 0.9630557  | 0.0074892 | -4.84 | 0.000 | 0.9479018 - 0.9784518 |
| weight    | 1.053197   | 0.003579  | 15.25 | 0.000 | 1.045923 - 1.060522  |
| age       | 1.066112   | 0.0039457 | 19.81 | 0.000 | 1.059107 - 1.073163  |
| _cons     | 0.3372393  | 0.4045108 | -0.91 | 0.372 | 0.029208 - 3.893807  |

Note: _cons estimates baseline odds.

Because the odds ratio for the `age` predictor is significantly greater than 1, we can conclude that older females are more likely to have high blood pressure than are younger females.
Health surveys

There are many sources of bias when modeling the association between a disease and its risk factors (Korn, Graubard, and Midthune 1997; Korn and Graubard 1999, sec. 3.7). In cross-sectional health surveys, inference is typically restricted to the target population as it stood when the data were collected. This type of survey cannot capture the fact that participants may change their habits over time. Some health surveys collect data retrospectively, relying on the participants to recall the status of risk factors as they stood in the past. This type of survey is vulnerable to recall bias.

Longitudinal surveys collect data over time, monitoring the survey participants over several years. Although the above biases are minimized, analysts are still faced with some tough choices/situations when modeling time-to-event data. For example:

- Time scale. When studying cancer, should we measure the time scale by using the participant’s age or the initial date from which data were collected?
- Time-varying covariates. Were all relevant risk factors sampled over time, or do we have only the baseline measurement?
- Competing risks. When studying mortality, do we have the data specific to cause of death?


Example 3: Cox’s proportional hazards model

Suppose that we want to model the incidence of lung cancer by using three risk factors: smoking status, sex, and place of residence. Our dataset comes from a longitudinal health survey: the First National Health and Nutrition Examination Survey (NHANES I) (Miller 1973; Engel et al. 1978) and its 1992 Epidemiologic Follow-up Study (NHEFS) (Cox et al. 1997); see the National Center for Health Statistics website at https://www.cdc.gov/nchs/. We will be using data from the samples identified by NHANES I examination locations 1–65 and 66–100; thus we will `svyset` the revised pseudo-PSU and strata variables associated with these locations. Similarly, our `pweight` variable was generated using the sampling weights for the nutrition and detailed samples for locations 1–65 and the weights for the detailed sample for locations 66–100.

```
use https://www.stata-press.com/data/r16/nhefs
svyset psu2 [pw=swgt2], strata(strata2)
pweight: swgt2
VCE: linearized
Single unit: missing
Strata 1: strata2
SU 1: psu2
FPC 1: <zero>
```

The lung cancer information was taken from the 1992 NHEFS interview data. We use the participants’ ages for the time scale. Participants who never had lung cancer and were alive for the 1992 interview were considered censored. Participants who never had lung cancer and died before the 1992 interview were also considered censored at their age of death.
10 svy estimation — Estimation commands for survey data

```
. stset age_lung_cancer [pw=swgt2], fail(lung_cancer)
  failure event: lung_cancer != 0 & lung_cancer < .
  obs. time interval: (0, age_lung_cancer]
  exit on or before: failure
  weight: [pweight=swgt2]

14,407 total observations
5,126 event time missing (age_lung_cancer>=.) PROBABLE ERROR
9,281 observations remaining, representing
83 failures in single-record/single-failure data
599,691 total analysis time at risk and under observation
  at risk from t = 0
  earliest observed entry t = 0
  last observed exit t = 97
```

Although `stset` warns us that it is a “probable error” to have 5,126 observations with missing event times, we can verify from the 1992 NHEFS documentation that there were indeed 9,281 participants with complete information.

For our proportional hazards model, we pulled the risk factor information from the NHANES I and 1992 NHEFS datasets. Smoking status was taken from the 1992 NHEFS interview data, but we filled in all but 132 missing values by using the general medical history supplement data in NHANES I. Smoking status is represented by separate indicator variables for former smokers and current smokers; the base comparison group is nonsmokers. Sex was determined using the 1992 NHEFS vitality data and is represented by an indicator variable for males. Place-of-residence information was taken from the medical history questionnaire in NHANES I and is represented by separate indicator variables for rural and heavily populated (more than 1 million people) urban residences; the base comparison group is urban residences with populations of fewer than 1 million people.

```
. svy: stcox former_smoker smoker male urban1 rural
  (running stcox on estimation sample)
Survey: Cox regression
Number of strata   =      35                        Number of obs   =     9,149
Number of PSUs    =     105                        Population size =   151,327,827
Design df         =     70                          F(  5,  66)    =     14.07
Prob > F          =   0.0000
```

<table>
<thead>
<tr>
<th></th>
<th>Linearized</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Haz. Ratio</td>
</tr>
<tr>
<td>former_smoker</td>
<td>2.788113</td>
</tr>
<tr>
<td>smoker</td>
<td>7.849483</td>
</tr>
<tr>
<td>male</td>
<td>1.187611</td>
</tr>
<tr>
<td>urban1</td>
<td>.8035074</td>
</tr>
<tr>
<td>rural</td>
<td>1.581674</td>
</tr>
</tbody>
</table>

From the above results, we can see that both former and current smokers have a significantly higher risk for developing lung cancer than do nonsmokers.
Technical note

In the previous example, we specified a sampling weight variable in the calls to both svyset and stset. When the svy prefix is used with stcox and streg, it identifies the sampling weight variable by using the data characteristics from both svyset and stset. svy will report an error if the svyset pweight variable is different from the stset pweight variable. The svy prefix will use the specified pweight variable, even if it is svyset but not stset. If a pweight variable is stset but not svyset, svy will note that it will be using the stset pweight variable and then svyset it.

The standard st commands will not use the svyset pweight variable if it is not also stset.

Example 4: Multiple baseline hazards

We can assess the proportional-hazards assumption across the observed race categories for the model fit in the previous example. The race information in our 1992 NHEFS dataset is contained in the revised_race variable. We will use stphplot to produce a log-log plot for each category of revised_race. As described in [ST] stcox PH-assumption tests, if the plotted lines are reasonably parallel, the proportional-hazards assumption has not been violated. We will use the zero option to reset the risk factors to their base comparison group.

```
  . stphplot, strata(revised_race) adjust(former_smoker smoker male urban1 rural)
     > zero legend(col(1))
       failure _d: lung_cancer
       analysis time _t: age_lung_cancer
       weight: [pweight=swgt2]
```

As we can see from the graph produced above, the lines for the black and white race categories intersect. This indicates a violation of the proportional-hazards assumption, so we should consider using separate baseline hazard functions for each race category in our model fit. We do this next, by specifying strata(revised_race) in our call to svy: stcox.
. svy: stcox former_smoker smoker male urban1 rural, strata(revised_race)
(running stcox on estimation sample)
Survey: Cox regression

<table>
<thead>
<tr>
<th></th>
<th>Linearized</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Haz. Ratio</td>
<td>Std. Err.</td>
<td>t</td>
<td>P&gt;</td>
<td>t</td>
<td></td>
</tr>
<tr>
<td>former_smoker</td>
<td>2.801797</td>
<td>.6280352</td>
<td>4.60</td>
<td>0.000</td>
<td>1.791761</td>
<td>4.381201</td>
</tr>
<tr>
<td>smoker</td>
<td>7.954921</td>
<td>2.640022</td>
<td>6.25</td>
<td>0.000</td>
<td>4.103709</td>
<td>15.42038</td>
</tr>
<tr>
<td>male</td>
<td>1.165724</td>
<td>.3390339</td>
<td>0.53</td>
<td>0.600</td>
<td>.6526527</td>
<td>2.082139</td>
</tr>
<tr>
<td>urban1</td>
<td>.784031</td>
<td>.3120525</td>
<td>-0.61</td>
<td>0.543</td>
<td>.3544764</td>
<td>1.73412</td>
</tr>
<tr>
<td>rural</td>
<td>1.490269</td>
<td>.5048569</td>
<td>1.18</td>
<td>0.243</td>
<td>.7582848</td>
<td>2.928851</td>
</tr>
</tbody>
</table>

Stratified by revised_race

References


Also see

[SVY] svy postestimation — Postestimation tools for svy

[SVY] estat — Postestimation statistics for survey data

[SVY] svy — The survey prefix command

[SVY] svyset — Declare survey design for dataset

[SVY] Calibration — Calibration for survey data

[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] Variance estimation — Variance estimation for survey data

[U] 20 Estimation and postestimation commands