

Postestimation commands

Remarks and examples

predict

References

margins

Also see

## Postestimation commands

The following postestimation commands are available after `svy`:

command	Description
<code>contrast</code>	contrasts and ANOVA-style joint tests of parameters
<code>estat (svy)</code>	postestimation statistics for survey data
<code>estimates</code>	cataloging estimation results
<code>etable</code>	table of estimation results
<code>lincom</code>	point estimates, standard errors, testing, and inference for linear combinations of parameters
<code>margins</code>	marginal means, predictive margins, marginal effects, and average marginal effects
<code>marginsplot</code>	graph the results from margins (profile plots, interaction plots, etc.)
<code>nlcom</code>	point estimates, standard errors, testing, and inference for nonlinear combinations of parameters
<code>predict</code>	predictions, residuals, influence statistics, and other diagnostic measures
<code>predictnl</code>	point estimates, standard errors, testing, and inference for generalized predictions
<code>pwcompare</code>	pairwise comparisons of parameters
<code>suest</code>	seemingly unrelated estimation
<code>test</code>	Wald tests of simple and composite linear hypotheses
<code>testnl</code>	Wald tests of nonlinear hypotheses

See [SVY] [estat](#).

## predict

The syntax of `predict` (and even if `predict` is allowed) after `svy` depends on the command used with `svy`. Specifically, `predict` is not allowed after `svy: mean`, `svy: proportion`, `svy: ratio`, `svy: tabulate`, or `svy: total`.

## margins

The syntax of `margins` (and even if `margins` is allowed) after `svy` depends on the command used with `svy`. Specifically, `margins` is not allowed after `svy: mean`, `svy: proportion`, `svy: ratio`, `svy: tabulate`, or `svy: total`.

## Remarks and examples

What follows are some examples of applications of postestimation commands using survey data. The examples are meant only to introduce the commands in a survey context and explore a few of the possibilities for postestimation analysis. See the individual entries for each command in the *Base Reference Manual* for complete syntax and many more examples.

### ► Example 1: Linear and nonlinear combinations

`lincom` will display an estimate of a linear combination of parameters, along with its standard error, a confidence interval, and a test that the linear combination is zero. `nlcom` will do likewise for nonlinear combinations of parameters.

`lincom` is commonly used to compute the differences of two subpopulation means. For example, suppose that we wish to estimate the difference of zinc levels in white males versus black males in the population represented by the NHANES II data (McDowell et al. 1981). Because the survey design characteristics are already `svyset` in `nhanes2.dta`, we only need to generate a variable for identifying the male subpopulation before using `svy: mean`.

```
. use https://www.stata-press.com/data/r19/nhanes2
. generate male = (sex == 1)
. svy, subpop(male): mean zinc, over(race)
(running mean on estimation sample)

Survey: Mean estimation
Number of strata = 31      Number of obs   =      9,811
Number of PSUs   = 62      Population size = 111,127,314
                          Subpop. no. obs   =      4,375
                          Subpop. size      = 50,129,281
                          Design df         =       31
```

	Linearized			
	Mean	std. err.	[95% conf. interval]	
c.zinc@race				
White	91.15725	.541625	90.0526	92.2619
Black	88.269	1.208336	85.80458	90.73342
Other	85.54716	2.608974	80.22612	90.8682

Then we run `lincom` to estimate the difference of zinc levels between the two subpopulations.

```
. lincom zinc#1.race - zinc#2.race
( 1)  c.zinc@1bn.race - c.zinc@2.race = 0
```

Mean	Coefficient	Std. err.	t	P> t	[95% conf. interval]	
(1)	2.888249	1.103999	2.62	0.014	.6366288	5.139868

The *t* statistic and its *p*-value give a survey analysis equivalent of a two-sample *t* test.

`lincom` and `nlcom` can be used after any of the estimation commands described in [SVY] **svy estimation**. `lincom` can, for example, display results as odds ratios after `svy: logit` and can be used to compute odds ratios for one covariate group relative to another. `nlcom` can display odds ratios, as well, and allows more general nonlinear combinations of the parameters. See [R] **lincom** and [R] **nlcom** for full details. Also see Eltinge and Sribney (1996) for an earlier implementation of `lincom` for survey data.

Finally, `lincom` and `nlcom` operate on the estimated parameters only. To obtain estimates and inference for functions of the parameters and of the data, such as for an exponentiated linear predictor or a predicted probability of success from a logit model, use `predictnl`; see [R] **predictnl**.

## ► Example 2: Quadratic terms

From [example 2](#) in [\[SVY\] svy estimation](#), we modeled the incidence of high blood pressure as a function of height, weight, age, and sex (using the female indicator variable). Here we also include  $c.age\#c.age$ , a squared term for age.

```
. use https://www.stata-press.com/data/r19/nhanes2d, clear
. svy: logistic highbp height weight age c.age#c.age female
(running logistic on estimation sample)

Survey: Logistic regression

Number of strata = 31                Number of obs   =      10,351
Number of PSUs   = 62                Population size = 117,157,513
                                          Design df       =         31
                                          F(5, 27)        =      284.33
                                          Prob > F         =      0.0000
```

highbp	Odds ratio	Linearized std. err.	t	P> t	[95% conf. interval]	
height	.9656421	.005163	-6.54	0.000	.9551693	.9762298
weight	1.052911	.0026433	20.54	0.000	1.047534	1.058316
age	1.055829	.0133575	4.29	0.000	1.028935	1.083426
c.age#c.age	.9999399	.0001379	-0.44	0.666	.9996588	1.000221
female	.6260988	.0364945	-8.03	0.000	.5559217	.7051347
_cons	.653647	.5744564	-0.48	0.632	.108869	3.924483

Note: **\_cons** estimates baseline odds.

Because our model includes a quadratic in the age variable, the peak incidence of high blood pressure with respect to age will occur at  $-\_b[age]/(2*\_b[c.age\#c.age])$ , which we can estimate, along with its standard error, using `nlcom`.

```
. nlcom peak: -_b[age]/(2*_b[c.age#c.age])
      peak: -_b[age]/(2*_b[c.age#c.age])
```

highbp	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
peak	452.0979	933.512	0.48	0.628	-1377.552	2281.748

Or we can use `testnl` to test that the peak incidence of high blood pressure in the population is 70 years.

```
. testnl -_b[age]/(2*_b[c.age#c.age]) = 70
      (1)  -_b[age]/(2*_b[c.age#c.age]) = 70
              chi2(1) =      0.17
              Prob > chi2 =    0.6823
```

These data do not reject our theory. `testnl` allows multiple hypotheses to be tested jointly and applies the degrees-of-freedom adjustment for survey results; see [\[R\] testnl](#).

### ► Example 3: Predictive margins

Changing our logistic regression for high blood pressure slightly, we add a factor variable for the levels of race. Level 1 of race represents whites, level 2 represents blacks, and level 3 represents others. We also specify that female is a factor variable, which does not change its coefficient but does increase its functionality with some postestimation commands.

```
. svy: logistic highbp height weight age c.age#c.age i.female i.race, baselevels
(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(7, 25)        =      230.16
```

```
Prob > F        =       0.0000
```

highbp	Odds ratio	Linearized std. err.	t	P> t	[95% conf. interval]	
height	.9675961	.0052361	-6.09	0.000	.9569758	.9783343
weight	1.052683	.0026091	20.72	0.000	1.047376	1.058018
age	1.056628	.0134451	4.33	0.000	1.029559	1.084408
c.age#c.age	.9999402	.0001382	-0.43	0.668	.9996585	1.000222
female						
Male	1	(base)				
Female	.6382331	.0377648	-7.59	0.000	.5656774	.720095
race						
White	1	(base)				
Black	1.422003	.1556023	3.22	0.003	1.137569	1.777557
Other	1.63456	.2929919	2.74	0.010	1.13405	2.355971
_cons	.4312846	.378572	-0.96	0.345	.0719901	2.583777

Note: **\_cons** estimates baseline odds.

Our point estimates indicate that the odds of females having high blood pressure is about 64% of the odds for men and that the odds of blacks having high blood pressure is about 1.4 times that of whites. The odds ratios give us the relative effects of their covariates, but they do not give us any sense of the absolute size of the effects. The odds ratio comparing blacks with whites is clearly large and statistically significant, but does it represent a sizable change? One way to answer that question is to explore the probabilities of high blood pressure from our fitted model. Let's first look at the predictive margins of the probability of high blood pressure for the three levels of race.

```
. margins race, vce(unconditional)
```

```
Predictive margins
```

```
Number of strata = 31
```

```
Number of PSUs   = 62
```

```
Number of obs    =    10,351
```

```
Population size  = 117,157,513
```

```
Design df       =         31
```

```
Expression: Pr(highbp), predict()
```

	Margin	Linearized std. err.	t	P> t	[95% conf. interval]	
race						
White	.3600722	.0150121	23.99	0.000	.3294548	.3906895
Black	.4256413	.0211311	20.14	0.000	.3825441	.4687385
Other	.4523404	.0311137	14.54	0.000	.3888836	.5157972

Because our response is a probability, these margins are sometimes called predicted marginal proportions or model-adjusted risks. They let us compare the effect of our three racial groups while controlling for the distribution of other covariates in the groups. Computationally, these predictive margins are the weighted average of the predicted probabilities for each observation in the estimation sample. The marginal probability for whites is the average probability, assuming that everyone in the sample is white; the margin for blacks assumes that everyone is black; and the margin for others assumes that everyone is something other than black or white.

There is a sizable difference in blood pressure between whites and blacks, with the marginal probability of high blood pressure for whites being about 36% and that for blacks being almost 43%. These are the adjusted probability levels. A more direct answer to our question about whether the odds ratios represent a substantial effect requires looking at the differences of these marginal probabilities. Researchers in the health-related sciences call such differences risk differences, whereas researchers in the social sciences usually call them average marginal effects or average partial effects.

Regardless of terminology, we are interested in the difference in the probability of blacks having high blood pressure as compared with whites, while adjusting for all other covariates in the model. We request risk differences by specifying the variables of interest in a `dydx()` option.

```
. margins, vce(unconditional) dydx(race)
```

```
Average marginal effects
```

```
Number of strata = 31
```

```
Number of PSUs   = 62
```

```
Number of obs    =    10,351
```

```
Population size  = 117,157,513
```

```
Design df       =         31
```

```
Expression: Pr(highbp), predict()
```

```
dy/dx wrt:  2.race 3.race
```

	dy/dx	Linearized std. err.	t	P> t	[95% conf. interval]	
race						
Black	.0655691	.0204063	3.21	0.003	.0239501	.1071881
Other	.0922682	.0343809	2.68	0.012	.0221478	.1623886

Note: dy/dx for factor levels is the discrete change from the base level.

Looking in the column labeled dy/dx, we see that the risk difference between blacks and whites is about 6.6% (0.6557). That is a sizable as well as significant difference.

Because they are population-weighted averages over the whole sample, these margins are estimates of the population average risk differences. And because we specified the `vce(unconditional)` option, their standard errors and confidence intervals can be used to make inferences about the population average risk differences. See [Methods and formulas](#) in [\[R\] margins](#) for details.

We can also compute margins or risk differences for subpopulations. To compute risk differences for the four subpopulations that are the regions of the United States—Northeast, Midwest, South, and West—we add the `over(region)` option.

```
. margins, vce(unconditional) dydx(race) over(region)
Average marginal effects
Number of strata = 31                Number of obs =      10,351
Number of PSUs  = 62                Population size = 117,157,513
                                   Design df   =          31

Expression: Pr(highbpb), predict()
dy/dx wrt:  2.race 3.race
Over:       region
```

		Linearized dy/dx	std. err.	t	P> t	[95% conf. interval]	
1.race		(base outcome)					
2.race							
	region						
	NE	.0662951	.0207354	3.20	0.003	.0240051	.1085852
	MW	.065088	.0204357	3.19	0.003	.0234091	.106767
	S	.0663448	.0202173	3.28	0.003	.0251112	.1075783
	W	.0647221	.0203523	3.18	0.003	.0232134	.1062308
3.race							
	region						
	NE	.093168	.0343919	2.71	0.011	.0230253	.1633106
	MW	.091685	.034247	2.68	0.012	.0218379	.1615322
	S	.0932303	.0345933	2.70	0.011	.0226769	.1637837
	W	.0912062	.034322	2.66	0.012	.021206	.1612063

Note: dy/dx for factor levels is the discrete change from the base level.

The differences in the covariate distributions across the regions have little effect on the risk differences between blacks and whites, or between other races and whites.

Rather than explore the probabilities after logistic regression, we might have explored the hazards or mean survival times after fitting a survival model. See [\[R\] margins](#) for many more applications of `margins`.

◀

## ► Example 4: Predictive means with replication-based variance estimators

When performing estimations with linearized standard errors, we use the `vce(unconditional)` option to compute marginal effects so that we can use the results to make inferences on the population. `margins` with `vce(unconditional)` uses linearization to compute the unconditional variance of the marginal means.

The `vce(unconditional)` option, therefore, cannot be used when a different variance estimation method has been specified for the model. If you are using a replication-based method to estimate the variance in your model, you may want to use this method to perform the variance estimation for your margins as well. To do that, you can write a program that performs both your main estimation and the computation of your margins and use the replication method with your program.

Continuing with the [logistic example](#), we will see how to estimate the marginal means for race by using the jackknife variance estimator. The program below accepts an argument that contains the estimation command line. Notice that the program should accept the `if` qualifier and also weights. In addition, the `set buildfvinfo` on command is included so that margins checks for estimability. `buildfvinfo` is usually set on, but is set off because it increases the computation time when you use replication methods; thus you need to set it on. The option `post` of margins posts the results to `e(b)`, so they can be used by `svy jackknife`.

```

program mymargins, eclass
    version 19.5    // (or version 19 if you do not have StataNow)
    syntax anything [if] [iw pw]
    if "'weight'" != "" {
        local wgtexp "['weight' 'exp']"
    }
    set buildfvinfo on
    `anything' `if' `wgtexp'
    margins race, post
end

```

We can now type

```

. local mycmdline logistic highbp height weight age c.age#c.age i.race i.female
. quietly mymargins `mycmdline'
. svy jackknife _b: mymargins `mycmdline'
(running mymargins on estimation sample)
Jackknife replications (62): .....10.....20.....30.....40.....
> ..50.....60.. done

Predictive margins
Number of strata = 31                Number of obs   =      10,351
Number of PSUs   = 62                Population size = 117,157,513
                                   Replications    =       62
                                   Design df         =       31

```

	Coefficient	Jackknife std. err.	t	P> t	[95% conf. interval]	
race						
White	.3600722	.0150128	23.98	0.000	.3294534	.390691
Black	.4256413	.0211504	20.12	0.000	.3825048	.4687778
Other	.4523404	.0322488	14.03	0.000	.3865684	.5181124

You can see that now the jackknife standard errors are being reported.

### ► Example 5: Nonlinear predictions and their standard errors

Continuing with the NHANES II data, we fit a linear regression of log of blood lead level on age, age-squared, gender, race, and region.

```
. use https://www.stata-press.com/data/r19/nhanes2d
. svy: regress loglead age c.age#c.age i.female i.race i.region
(running regress on estimation sample)

Survey: Linear regression
Number of strata = 31
Number of PSUs  = 62
Number of obs   = 4,948
Population size = 56,405,414
Design df       = 31
F(8, 24)        = 156.24
Prob > F         = 0.0000
R-squared       = 0.2379
```

loglead	Linearized		t	P> t	[95% conf. interval]	
	Coefficient	std. err.				
age	.0158388	.0027352	5.79	0.000	.0102603	.0214173
c.age#c.age	-.0001464	.0000295	-4.96	0.000	-.0002066	-.0000862
female						
Female	-.3655338	.0116157	-31.47	0.000	-.3892242	-.3418434
race						
Black	.178402	.0314173	5.68	0.000	.114326	.242478
Other	-.0516952	.0402381	-1.28	0.208	-.1337614	.030371
region						
MW	-.02283	.0389823	-0.59	0.562	-.1023349	.0566749
S	-.1685453	.056004	-3.01	0.005	-.2827662	-.0543244
W	-.0362295	.0387508	-0.93	0.357	-.1152623	.0428032
_cons	2.440671	.0627987	38.86	0.000	2.312592	2.568749

Given that we modeled the natural log of the lead measurement, we can use `predictnl` to compute the exponentiated linear prediction (in the original units of the lead variable), along with its standard error.

```
. predictnl leadhat = exp(xb()) if e(sample), se(leadhat_se)
(5,403 missing values generated)
. sort lead leadhat
. generate showobs = inrange(_n,1,5) + inrange(_n,2501,2505) +
> inrange(_n,4945,4948)
```



```
. list lead leadhat leadhat_se age c.age#c.age if showobs, abbrev(10)
```

	lead	leadhat	leadhat_se	age	c.age# c.age
1.	2	9.419804	.5433255	29	841
2.	3	8.966098	.5301117	23	529
3.	3	9.046788	.5298448	24	576
4.	3	9.046788	.5298448	24	576
5.	3	9.27693	.5347956	27	729
2501.	13	16.88317	.7728783	37	1369
2502.	13	16.90057	2.296082	71	5041
2503.	13	16.90057	2.296082	71	5041
2504.	13	16.90237	1.501056	48	2304
2505.	13	16.90852	2.018708	60	3600
4945.	61	17.18581	2.052034	58	3364
4946.	64	15.08437	.647629	24	576
4947.	66	17.78698	1.641349	56	3136
4948.	80	16.85864	1.333927	42	1764

◀

## ► Example 6: Multiple-hypothesis testing

Joint-hypothesis tests can be performed after svy commands with the `test` command. Using the results from the regression model fit in the [previous example](#), we can use `test` to test the joint significance of `2.region`, `3.region`, and `4.region`. (`1.region` is the Northeast, `2.region` is the Midwest, `3.region` is the South, and `4.region` is the West.) We test the hypothesis that `2.region = 0`, `3.region = 0`, and `4.region = 0`.

```
. test 2.region 3.region 4.region
```

Adjusted Wald test

```
( 1) 2.region = 0
( 2) 3.region = 0
( 3) 4.region = 0
```

```
F( 3, 29) = 2.96
Prob > F = 0.0486
```

The `nosvyadjust` option on `test` produces an unadjusted Wald test.

```
. test 2.region 3.region 4.region, nosvyadjust
```

Unadjusted Wald test

```
( 1) 2.region = 0
( 2) 3.region = 0
( 3) 4.region = 0
```

```
F( 3, 31) = 3.17
Prob > F = 0.0382
```

For one-dimensional tests, the adjusted and unadjusted  $F$  statistics are identical, but they differ for higher-dimensional tests. Using the `nosvyadjust` option is not recommended because the unadjusted  $F$  statistic can produce extremely anticonservative  $p$ -values (that is,  $p$ -values that are too small) when the variance degrees of freedom (equal to the number of sampled PSUs minus the number of strata) is not large relative to the dimension of the test.

Bonferroni-adjusted  $p$ -values can also be computed:

```
. test 2.region 3.region 4.region, mtest(bonferroni)
Adjusted Wald test
( 1) 2.region = 0
( 2) 3.region = 0
( 3) 4.region = 0
```

	F(df,29)	df	p > F
(1)	0.34	1	1.0000*
(2)	9.06	1	0.0155*
(3)	0.87	1	1.0000*
All	2.96	3	0.0486

\* Bonferroni-adjusted  $p$ -values

See [Korn and Graubard \(1990\)](#) for a discussion of these three different procedures for conducting joint-hypothesis tests. See [Eltinge and Sribney \(1996\)](#) for an earlier implementation of test for survey data.



## ► Example 7: Contrasts

After `svy` commands, we can estimate contrasts and make pairwise comparisons with the `contrast` and `pwcompare` commands. First, we will fit a regression of serum zinc levels on health status:

```
. use https://www.stata-press.com/data/r19/nhanes2f, clear
. label list hlthgrp
hlthgrp:
    1 Poor
    2 Fair
    3 Average
    4 Good
    5 Excellent
. svy: regress zinc i.health
(running regress on estimation sample)

Survey: Linear regression

Number of strata = 31          Number of obs =      9,188
Number of PSUs  = 62          Population size = 104,162,204
                                Design df      =      31
                                F(4, 28)        =     15.61
                                Prob > F         =     0.0000
                                R-squared        =     0.0098
```

zinc	Linearized		t	P> t	[95% conf. interval]	
	Coefficient	std. err.				
health						
Fair	.9272308	.7690396	1.21	0.237	-.6412357	2.495697
Average	2.444004	.6407097	3.81	0.001	1.137268	3.75074
Good	4.038285	.6830349	5.91	0.000	2.645226	5.431344
Excellent	4.770911	.7151641	6.67	0.000	3.312324	6.229498
_cons	83.94729	.8523379	98.49	0.000	82.20893	85.68564

Higher levels of zinc are associated with better health. We can use reverse adjacent contrasts to compare each health status with the preceding status.

```
. contrast ar.health
Contrasts of marginal linear predictions
```

Design df = 31

Margins: asbalanced

	df	F	P>F
health			
(Fair vs Poor)	1	1.45	0.2371
(Average vs Fair)	1	5.49	0.0257
(Good vs Average)	1	10.92	0.0024
(Excellent vs Good)	1	1.93	0.1744
Joint	4	15.61	0.0000
Design	31		

Note: F statistics are adjusted for the survey design.

	Contrast	Std. err.	[95% conf. interval]	
health				
(Fair vs Poor)	.9272308	.7690396	-.6412357	2.495697
(Average vs Fair)	1.516773	.6474771	.1962347	2.837311
(Good vs Average)	1.594281	.4824634	.6102904	2.578271
(Excellent vs Good)	.7326264	.5270869	-.3423744	1.807627

The first table reports significance tests for each contrast, along with a joint test of all the contrasts. The row labeled (Fair vs Poor), for example, tests the null hypothesis that the first two health statuses have the same mean zinc level. The test statistics are automatically adjusted for the survey design.

The second table reports estimates, standard errors, and confidence limits for each contrast. The row labeled (Good vs Average), for example, shows that those in good health have a mean zinc level about 1.6 units higher than those of average health. The standard errors and confidence intervals also account for the survey design.

If we would like to go further and make all possible pairwise comparisons of the health groups, we can use the `pwcompare` command. We will specify the `mcompare(sidak)` option to account for multiple comparisons and the `cformat(%3.1f)` option to reduce the number of decimal places in the output:

```
. pwcompare health, mcompare(sidak) cformat(%3.1f)
Pairwise comparisons of marginal linear predictions
```

Design df = 31

Margins: asbalanced

	Number of comparisons
health	10

	Contrast	Std. err.	Sidak [95% conf. interval]	
health				
Fair vs Poor	0.9	0.8	-1.4	3.2
Average vs Poor	2.4	0.6	0.5	4.4
Good vs Poor	4.0	0.7	2.0	6.1
Excellent vs Poor	4.8	0.7	2.6	6.9
Average vs Fair	1.5	0.6	-0.4	3.5
Good vs Fair	3.1	0.5	1.5	4.8
Excellent vs Fair	3.8	0.7	1.7	6.0
Good vs Average	1.6	0.5	0.1	3.0
Excellent vs Average	2.3	0.7	0.3	4.4
Excellent vs Good	0.7	0.5	-0.9	2.3

Seven of the ten Šidák intervals exclude the null value of zero. See [\[R\] pwcompare](#) for more information on pairwise comparisons and multiple-comparison adjustments.



### ► Example 8: Using `suest` with survey data, the `svy` prefix

`suest` can be used to obtain the variance estimates for a series of estimators that used the `svy` prefix. To use `suest` for this purpose, perform the following steps:

1. Be sure to set the survey design characteristics correctly by using `svyset`. Do not use the `vce()` option to change the default variance estimator from the linearized variance estimator. `vce(brr)` and `vce(jackknife)` are not supported by `suest`.
2. Fit the model or models by using the `svy` prefix command, optionally including subpopulation estimation with the `subpop()` option.
3. Store the estimation results with `estimates` store *name*.

In the following, we illustrate how to use `suest` to compare the parameter estimates between two ordered logistic regression models.

In the NHANES II dataset, we have the variable `health` containing self-reported health status, which takes on the values 1–5, with 1 being “poor” and 5 being “excellent”. Because this is an ordered categorical variable, it makes sense to model it by using `svy: ologit`. We use some basic demographic variables as predictors: `female` (an indicator of female individuals), `black` (an indicator for black individuals), `age` in years, and `c.age#c.age` (age squared).

```
. use https://www.stata-press.com/data/r19/nhanes2f, clear
. svyset psuid [pw=finalwgt], strata(stratid)
Sampling weights: finalwgt
                  VCE: linearized
    Single unit: missing
      Strata 1: stratid
Sampling unit 1: psuid
      FPC 1: <zero>

. svy: ologit health female black age c.age#c.age
(running ologit on estimation sample)

Survey: Ordered logistic regression

Number of strata = 31                Number of obs   =      10,335
Number of PSUs   = 62                Population size = 116,997,257
                                   Design df          =        31
                                   F(4, 28)            =      223.27
                                   Prob > F             =      0.0000
```

health	Linearized		t	P> t	[95% conf. interval]	
	Coefficient	std. err.				
female	-.1615219	.0523678	-3.08	0.004	-.2683267	-.054717
black	-.986568	.0790277	-12.48	0.000	-1.147746	-.8253899
age	-.0119491	.0082974	-1.44	0.160	-.0288717	.0049736
c.age#c.age	-.0003234	.000091	-3.55	0.001	-.000509	-.0001377
/cut1	-4.566229	.1632561			-4.899192	-4.233266
/cut2	-3.057415	.1699944			-3.404121	-2.710709
/cut3	-1.520596	.1714342			-1.870239	-1.170954
/cut4	-.242785	.1703965			-.590311	.104741

The self-reported `health` variable takes five categories. Categories 1 and 2 denote negative categories, whereas categories 4 and 5 denote positive categories. We wonder whether the distinctions between the two positive categories and between the two negative categories are produced in accordance with one latent dimension, which is an assumption of the ordered logistic model. To test one-dimensionality, we will collapse the five-point health measure into a three-point measure, refit the ordered logistic model, and compare the regression coefficients and cutpoints between the two analyses. If the single latent variable assumption is valid, the coefficients and cutpoints should match. This can be seen as a Hausman-style specification test. Estimation of the ordered logistic model parameters for survey data is by maximum pseudolikelihood. Neither estimator is fully efficient, and thus the assumptions for the classic Hausman test and for the `hausman` command are not satisfied. With `suest`, we can obtain an appropriate Hausman test for survey data.

To perform the Hausman test, we are already almost halfway there by following steps 1 and 2 for one of the models. We just need to store the current estimation results before moving on to the next model. Here we store the results with `estimates store` under the name H5, indicating that in this analysis, the dependent variable `health` has five categories.

```
. estimates store H5
```

We proceed by generating a new dependent variable `health3`, which maps values 1 and 2 into 2, 3 into 3, and 4 and 5 into 4. This transformation is conveniently accomplished with the `clip()` function. We then fit an `ologit` model with this new dependent variable and store the estimation results under the name H3.

```
. generate health3 = clip(health, 2, 4)
(2 missing values generated)
. svy: ologit health3 female black age c.age#c.age
(running ologit on estimation sample)
```

Survey: Ordered logistic regression

Number of strata = 31  
Number of PSUs = 62

Number of obs = 10,335  
Population size = 116,997,257  
Design df = 31  
F(4, 28) = 197.08  
Prob > F = 0.0000

health3	Linearized		t	P> t	[95% conf. interval]	
	Coefficient	std. err.				
female	-.1551238	.0563809	-2.75	0.010	-.2701133	-.0401342
black	-1.046316	.0728274	-14.37	0.000	-1.194849	-.8977836
age	-.0365408	.0073653	-4.96	0.000	-.0515624	-.0215192
c.age#c.age	-.00009	.0000791	-1.14	0.264	-.0002512	.0000713
/cut1	-3.655498	.1610211			-3.983903	-3.327093
/cut2	-2.109584	.1597057			-2.435306	-1.783862

```
. estimates store H3
```

We can now obtain the combined estimation results of the two models stored under H5 and H3 with design-based standard errors.

```
. suest H5 H3
```

```
Simultaneous survey results for H5, H3
```

```
Number of strata = 31
```

```
Number of PSUs = 62
```

```
Number of obs = 10,335
```

```
Population size = 116,997,257
```

```
Design df = 31
```

	Linearized					
	Coefficient	std. err.	t	P> t	[95% conf. interval]	
H5_health						
female	-.1615219	.0523678	-3.08	0.004	-.2683267	-.054717
black	-.986568	.0790277	-12.48	0.000	-1.147746	-.8253899
age	-.0119491	.0082974	-1.44	0.160	-.0288717	.0049736
c.age#c.age	-.0003234	.000091	-3.55	0.001	-.000509	-.0001377
/H5						
cut1	-4.566229	.1632561			-4.899192	-4.233266
cut2	-3.057415	.1699944			-3.404121	-2.710709
cut3	-1.520596	.1714342			-1.870239	-1.170954
cut4	-.242785	.1703965			-.590311	.104741
H3_health3						
female	-.1551238	.0563809	-2.75	0.010	-.2701133	-.0401342
black	-1.046316	.0728274	-14.37	0.000	-1.194849	-.8977836
age	-.0365408	.0073653	-4.96	0.000	-.0515624	-.0215192
c.age#c.age	-.00009	.0000791	-1.14	0.264	-.0002512	.0000713
/H3						
cut1	-3.655498	.1610211			-3.983903	-3.327093
cut2	-2.109584	.1597057			-2.435306	-1.783862

The coefficients of H3 and H5 look rather similar. We now use `test` to perform a formal Hausman-type test for the hypothesis that the regression coefficients are indeed the same, as we would expect if there is indeed a one-dimensional latent dimension for health. Thus we test that the coefficients in the equation H5\_health are equal to those in H3\_health3.

```
. test [H5_health=H3_health3]
```

```
Adjusted Wald test
```

```
( 1) [H5_health]female - [H3_health3]female = 0
```

```
( 2) [H5_health]black - [H3_health3]black = 0
```

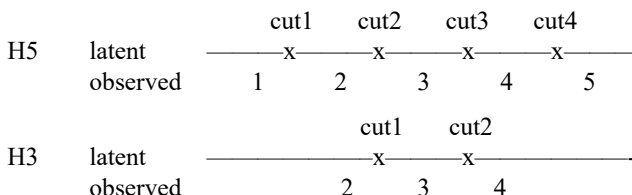
```
( 3) [H5_health]age - [H3_health3]age = 0
```

```
( 4) [H5_health]c.age#c.age - [H3_health3]c.age#c.age = 0
```

```
F( 4, 28) = 17.13
```

```
Prob > F = 0.0000
```

We can reject the null hypothesis, which indicates that the ordered logistic regression model is indeed misspecified. Another specification test can be conducted with respect to the cutpoints. Variable `health3` was constructed from `health` by collapsing the two worst categories into value 2 and the two best categories into value 4. This action effectively has removed two cutpoints, but if the model fits the data, it should not affect the other two cutpoints. The comparison is hampered by a difference in the names of the cutpoints between the models, as illustrated in the figure below:



Cutpoint /cut2 of model H5 should be compared with cutpoint /cut1 of H3, and similarly, /cut3 of H5 with /cut2 of H3.

```
. test (/H5:cut2=/H3:cut1) (/H5:cut3=/H3:cut2)
Adjusted Wald test
( 1)  [/H5]cut2 - [/H3]cut1 = 0
( 2)  [/H5]cut3 - [/H3]cut2 = 0
      F( 2, 30) = 33.49
      Prob > F = 0.0000
```

We conclude that the invariance of the cutpoints under the collapse of categories is not supported by the data, again providing evidence against the reduced specification of the ordered logistic model in this case.



### ► Example 9: Using `suest` with survey data, the `svy` option

Not all estimation commands support the `svy` prefix, but you can use the `svy` option with `suest` to get survey estimation results. If you can use `suest` after a command, you can use `suest, svy`. Here are the corresponding Stata commands to perform the analysis in the [previous example](#), using the `svy` option instead of the `svy` prefix.

```
. use https://www.stata-press.com/data/r19/nhanes2f, clear
. svyset psuid [pw=finalwgt], strata(stratid)
. ologit health female black age c.age#c.age [iw=finalwgt]
. estimates store H5
. generate health3 = clip(health,2,4)
. ologit health3 female black age c.age#c.age [iw=finalwgt]
. estimates store H3
. suest H5 H3, svy
. test [H5_health=H3_health3]
. test (/H5:cut2=/H3:cut1) (/H5:cut3=/H3:cut2)
```

The calls to `ologit` now use `iweights` instead of the `svy` prefix, and the `svy` option was added to `suest`. No other changes are required.





## References

- Eltinge, J. L., and W. M. Sribney. 1996. **svy5: Estimates of linear combinations and hypothesis tests for survey data**. *Stata Technical Bulletin* 31: 31–42. Reprinted in *Stata Technical Bulletin Reprints*, vol. 6, pp. 246–259. College Station, TX: Stata Press.
- Graubard, B. I., and E. L. Korn. 2004. Predictive margins with survey data. *Biometrics* 55: 652–659. <https://doi.org/10.1111/j.0006-341X.1999.00652.x>.
- Korn, E. L., and B. I. Graubard. 1990. Simultaneous testing of regression coefficients with complex survey data: Use of Bonferroni *t* statistics. *American Statistician* 44: 270–276. <https://doi.org/10.2307/2684345>.
- McDowell, A., A. Engel, J. T. Massey, and K. Maurer. 1981. “Plan and operation of the Second National Health and Nutrition Examination Survey, 1976–1980”. In *Vital and Health Statistics*, ser. 1, no. 15. Hyattsville, MD: National Center for Health Statistics.

## Also see

- [SVY] **svy** — The survey prefix command
- [SVY] **estat** — Postestimation statistics for survey data
- [SVY] **svy bootstrap** — Bootstrap for survey data
- [SVY] **svy brr** — Balanced repeated replication for survey data
- [SVY] **svy estimation** — Estimation commands for survey data
- [SVY] **svy jackknife** — Jackknife estimation for survey data
- [SVY] **svy sdr** — Successive difference replication for survey data
- [U] **13.5 Accessing coefficients and standard errors**
- [U] **20 Estimation and postestimation commands**

Stata, Stata Press, and Mata are registered trademarks of StataCorp LLC. Stata and Stata Press are registered trademarks with the World Intellectual Property Organization of the United Nations. StataNow and NetCourseNow are trademarks 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](#).

