

**vwls** — Variance-weighted least squares

<a href="#">Description</a>	<a href="#">Quick start</a>	<a href="#">Menu</a>	<a href="#">Syntax</a>
<a href="#">Options</a>	<a href="#">Remarks and examples</a>	<a href="#">Stored results</a>	<a href="#">Methods and formulas</a>
<a href="#">References</a>	<a href="#">Also see</a>		

## Description

`vwls` estimates a linear regression using variance-weighted least squares. It differs from ordinary least-squares (OLS) regression in that it does not assume homogeneity of variance, but requires that the conditional variance of *depvar* be estimated prior to the regression. The estimated variance need not be constant across observations. `vwls` treats the estimated variance as if it were the true variance when it computes standard errors of the coefficients.

You must supply an estimate of the conditional standard deviation of *depvar* to `vwls` by using the `sd(varname)` option, or you must have grouped data with the groups defined by the *indepvars* variables. In the latter case, `vwls` treats all *indepvars* as categorical variables, computes the mean and standard deviation of *depvar* separately for each subgroup, and computes the regression of the subgroup means on *indepvars*.

`regress` with analytic weights can be used to produce another kind of “variance-weighted least squares”; see [Remarks and examples](#) for an explanation of the difference.

## Quick start

Variance-weighted least-squares regression of *y* on *x1* and *x2*, with the estimated conditional std. dev. of *y* stored in `sd`

```
vwls y1 x1 x2, sd(sd)
```

Add categorical variable *a* using [factor-variable](#) syntax

```
vwls y1 x1 x2 i.a, sd(sd)
```

Same as above, but restrict the sample to cases where *v* is greater than 1

```
vwls y1 x1 x2 i.a if v>1, sd(sd)
```

Variance-weighted least-squares regression for grouped data with subgroups defined by *a2* and *a3*

```
vwls y2 i.a2 i.a3
```

## Menu

Statistics > Linear models and related > Other > Variance-weighted least squares

## Syntax

```
vwls depvar indepvars [if] [in] [weight] [, options]
```

<i>options</i>	Description
<b>Model</b>	
<code>noconstant</code>	suppress constant term
<code>sd(<i>varname</i>)</code>	variable containing estimate of conditional standard deviation
<b>Reporting</b>	
<code>level(#)</code>	set confidence level; default is <code>level(95)</code>
<code>display_options</code>	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
<code>coeflegend</code>	display legend instead of statistics
<i>indepvars</i> may contain factor variables; see [U] 11.4.3 <b>Factor variables</b> .	
<code>bootstrap</code> , <code>by</code> , <code>collect</code> , <code>jackknife</code> , <code>rolling</code> , and <code>statsby</code> are allowed; see [U] 11.1.10 <b>Prefix commands</b> .	
Weights are not allowed with the <code>bootstrap</code> prefix; see [R] <b>bootstrap</b> .	
<code>fweights</code> are allowed; see [U] 11.1.6 <b>weight</b> .	
<code>coeflegend</code> does not appear in the dialog box.	
See [U] 20 <b>Estimation and postestimation commands</b> for more capabilities of estimation commands.	

## Options

### Model

`noconstant`; see [R] **Estimation options**.

`sd(varname)` is an estimate of the conditional standard deviation of *depvar* (that is, it can vary observation by observation). All values of *varname* must be  $> 0$ . If you specify `sd()`, you cannot use `fweights`.

If `sd()` is not given, the data will be grouped by *indepvars*. Here *indepvars* are treated as categorical variables, and the means and standard deviations of *depvar* for each subgroup are calculated and used for the regression. Any subgroup for which the standard deviation is zero is dropped.

### Reporting

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

`display_options`: `noci`, `nopvalues`, `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(style)`, `cformat(%fmt)`, `pformat(%fmt)`, `sformat(%fmt)`, and `nolstretch`; see [R] **Estimation options**.

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

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

## Remarks and examples

The `vwls` command is intended for use with two special—and different—types of data. The first contains data that consist of measurements from physical science experiments in which all error is due solely to measurement errors and the sizes of the measurement errors are known.

You can also use variance-weighted least-squares linear regression for certain problems in categorical data analysis, such as when all the independent variables are categorical and the outcome variable is either continuous or a quantity that can sensibly be averaged. If each of the subgroups defined by the categorical variables contains a reasonable number of subjects, then the variance of the outcome variable can be estimated independently within each subgroup. For the purposes of estimation, `vwls` treats each subgroup as one observation, with the dependent variable being the subgroup mean of the outcome variable.

The `vwls` command fits the model

$$y_i = \mathbf{x}_i\boldsymbol{\beta} + \varepsilon_i$$

where the errors  $\varepsilon_i$  are independent normal random variables with the distribution  $\varepsilon_i \sim N(0, \nu_i)$ . The independent variables  $\mathbf{x}_i$  are assumed to be known without error.

As described above, `vwls` assumes that you already have estimates  $s_i^2$  for the variances  $\nu_i$ . The error variance is not estimated in the regression. The estimates  $s_i^2$  are used to compute the standard errors of the coefficients; see [Methods and formulas](#) below.

In contrast, weighted OLS regression assumes that the errors have the distribution  $\varepsilon_i \sim N(0, \sigma^2/w_i)$ , where the  $w_i$  are known weights and  $\sigma^2$  is an unknown parameter that is estimated in the regression. This is the difference from variance-weighted least squares: in weighted OLS, the magnitude of the error variance is estimated in the regression using all the data.

### ► Example 1

An artificial, but informative, example illustrates the difference between variance-weighted least squares and weighted OLS.

We measure the quantities  $x_i$  and  $y_i$  and estimate that the standard deviation of  $y_i$  is  $s_i$ . We enter the data into Stata:

```
. use https://www.stata-press.com/data/r18/vwlsxmpl
. list
```

	x	y	s
1.	1	1.2	.5
2.	2	1.9	.5
3.	3	3.2	1
4.	4	4.3	1
5.	5	4.9	1
6.	6	6.0	2
7.	7	7.2	2
8.	8	7.9	2

Because we want observations with smaller variance to carry larger weight in the regression, we compute an OLS regression with analytic weights proportional to the inverse of the squared standard deviations:

## 4 vwls — Variance-weighted least squares

```
. regress y x [aweight=s^(-2)]
(sum of wgt is 11.75)
```

Source	SS	df	MS	Number of obs	=	8
Model	22.6310183	1	22.6310183	F(1, 6)	=	702.26
Residual	.193355117	6	.032225853	Prob > F	=	0.0000
Total	22.8243734	7	3.26062477	R-squared	=	0.9915
				Adj R-squared	=	0.9901
				Root MSE	=	.17952

  

y	Coefficient	Std. err.	t	P> t	[95% conf. interval]
x	.9824683	.0370739	26.50	0.000	.8917517 1.073185
_cons	.1138554	.1120078	1.02	0.349	-.1602179 .3879288

If we compute a variance-weighted least-squares regression by using `vwls`, we get the same results for the coefficient estimates but very different standard errors:

```
. vwls y x, sd(s)
```

Variance-weighted least-squares regression	Number of obs	=	8
Goodness-of-fit chi2(6) = 0.28	Model chi2(1)	=	33.24
Prob > chi2 = 0.9996	Prob > chi2	=	0.0000

  

y	Coefficient	Std. err.	z	P> z	[95% conf. interval]
x	.9824683	.170409	5.77	0.000	.6484728 1.316464
_cons	.1138554	.51484	0.22	0.825	-.8952124 1.122923

Although the values of  $y_i$  were nicely linear with  $x_i$ , the `vwls` regression used the large estimates for the standard deviations to compute large standard errors for the coefficients. For weighted OLS regression, however, the scale of the analytic weights has no effect on the standard errors of the coefficients—only the relative proportions of the analytic weights affect the regression.

If we are sure of the sizes of our error estimates for  $y_i$ , using `vwls` is valid. However, if we can estimate only the relative proportions of error among the  $y_i$ , then `vwls` is not appropriate.

◀

### ► Example 2

Let's now consider an example of the use of `vwls` with categorical data. Suppose that we have blood pressure data for  $n = 400$  subjects, categorized by gender and race (black or white). Here is a description of the data:

```
. use https://www.stata-press.com/data/r18/bp
. table gender race, statistic(mean bp) statistic(sd bp) statistic(freq)
> nformat(%8.1f)
```

	Race		
	White	Black	Total
<b>Gender</b>			
<b>Female</b>			
Mean	117.1	118.5	117.8
Standard deviation	10.3	11.6	10.9
Frequency	100.0	100.0	200.0
<b>Male</b>			
Mean	122.1	125.8	124.0
Standard deviation	10.6	15.5	13.3
Frequency	100.0	100.0	200.0
<b>Total</b>			
Mean	119.6	122.2	120.9
Standard deviation	10.7	14.1	12.6
Frequency	200.0	200.0	400.0

Performing a variance-weighted regression using vwls gives

```
. vwls bp gender race
Variance-weighted least-squares regression      Number of obs      =      400
Goodness-of-fit chi2(1) = 0.88                 Model chi2(2)      =      27.11
Prob > chi2 = 0.3486                          Prob > chi2        =      0.0000
```

bp	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
gender	5.876522	1.170241	5.02	0.000	3.582892	8.170151
race	2.372818	1.191683	1.99	0.046	.0371631	4.708473
_cons	116.6486	.9296297	125.48	0.000	114.8266	118.4707

By comparison, an OLS regression gives the following result:

```
. regress bp gender race
Source |          SS          df          MS      Number of obs =      400
-----+-----+-----+-----+-----+-----
Model | 4485.66639          2    2242.83319    F(2, 397)      =      15.24
Residual | 58442.7305        397    147.210908    Prob > F        =      0.0000
Total | 62928.3969        399    157.71528    R-squared       =      0.0713
                                           Adj R-squared  =      0.0666
                                           Root MSE      =      12.133
```

bp	Coefficient	Std. err.	t	P> t	[95% conf. interval]	
gender	6.1775	1.213305	5.09	0.000	3.792194	8.562806
race	2.5875	1.213305	2.13	0.034	.2021938	4.972806
_cons	116.4862	1.050753	110.86	0.000	114.4205	118.552

Note the larger value for the race coefficient (and smaller  $p$ -value) in the OLS regression. The assumption of homogeneity of variance in OLS means that the mean for black men pulls the regression line higher than in the vwls regression, which takes into account the larger variance for black men and reduces its effect on the regression.

## Stored results

`vwls` stores the following in `e()`:

### Scalars

<code>e(N)</code>	number of observations
<code>e(df_m)</code>	model degrees of freedom
<code>e(chi2)</code>	model $\chi^2$
<code>e(df_gf)</code>	goodness-of-fit degrees of freedom
<code>e(chi2_gf)</code>	goodness-of-fit $\chi^2$
<code>e(rank)</code>	rank of <code>e(V)</code>

### Macros

<code>e(cmd)</code>	<code>vwls</code>
<code>e(cmdline)</code>	command as typed
<code>e(depvar)</code>	name of dependent variable
<code>e(properties)</code>	<code>b V</code>
<code>e(predict)</code>	program used to implement <code>predict</code>
<code>e(asbalanced)</code>	factor variables <code>fvset</code> as <code>asbalanced</code>
<code>e(asobserved)</code>	factor variables <code>fvset</code> as <code>asobserved</code>

### Matrices

<code>e(b)</code>	coefficient vector
<code>e(V)</code>	variance–covariance matrix of the estimators

### Functions

<code>e(sample)</code>	marks estimation sample
------------------------	-------------------------

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

### Matrices

<code>r(table)</code>	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

Let  $\mathbf{y} = (y_1, y_2, \dots, y_n)'$  be the vector of observations of the dependent variable, where  $n$  is the number of observations. When `sd()` is specified, let  $s_1, s_2, \dots, s_n$  be the standard deviations supplied by `sd()`. For categorical data, when `sd()` is not given, the means and standard deviations of  $y$  for each subgroup are computed, and  $n$  becomes the number of subgroups,  $\mathbf{y}$  is the vector of subgroup means, and  $s_i$  are the standard deviations for the subgroups.

Let  $\mathbf{V} = \text{diag}(s_1^2, s_2^2, \dots, s_n^2)$  denote the estimate of the variance of  $\mathbf{y}$ . Then the estimated regression coefficients are

$$\mathbf{b} = (\mathbf{X}'\mathbf{V}^{-1}\mathbf{X})^{-1}\mathbf{X}'\mathbf{V}^{-1}\mathbf{y}$$

and their estimated covariance matrix is

$$\widehat{\text{Cov}}(\mathbf{b}) = (\mathbf{X}'\mathbf{V}^{-1}\mathbf{X})^{-1}$$

A statistic for the goodness of fit of the model is

$$Q = (\mathbf{y} - \mathbf{X}\mathbf{b})' \mathbf{V}^{-1} (\mathbf{y} - \mathbf{X}\mathbf{b})$$

where  $Q$  has a  $\chi^2$  distribution with  $n - k$  degrees of freedom ( $k$  is the number of independent variables plus the constant, if any).

## References

- Gini, R., and J. Pasquini. 2006. [Automatic generation of documents](#). *Stata Journal* 6: 22–39.
- Grizzle, J. E., C. F. Starmer, and G. G. Koch. 1969. Analysis of categorical data by linear models. *Biometrics* 25: 489–504. <https://doi.org/10.2307/2528901>.

## Also see

- [R] [vwls postestimation](#) — Postestimation tools for vwls
- [R] [regress](#) — Linear regression
- [U] [11.1.6 weight](#)
- [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 © 1985–2023 StataCorp LLC, College Station, TX, USA. All rights reserved.



For suggested citations, see the FAQ on [citing Stata documentation](#).