

**mvreg** — Multivariate regression

[Description](#)  
[Options](#)  
[References](#)

[Quick start](#)  
[Remarks and examples](#)  
[Also see](#)

[Menu](#)  
[Stored results](#)

[Syntax](#)  
[Methods and formulas](#)

## Description

`mvreg` fits a multivariate regression model for several dependent variables with the same independent variables.

## Quick start

Multivariate regression of three `y` variables on two `x` variables

```
mvreg y1 y2 y3 = x1 x2
```

Add [indicator variables](#) for each level of categorical variable `catvar`

```
mvreg y1 y2 y3 = x1 x2 i.catvar
```

Model of three `y` variables on five `x` variables reporting the correlations between the residuals and the Breusch–Pagan test of independence

```
mvreg y1 y2 y3 = x1 x2 x3 x4 x5, corr
```

## Menu

Statistics > Multivariate analysis > MANOVA, multivariate regression, and related > Multivariate regression

## Syntax

```
mvreg depvars = indepvars [if] [in] [weight] [, options]
```

<i>options</i>	Description
<b>Model</b>	
<code>noconstant</code>	suppress constant term
<b>Reporting</b>	
<code>level(#)</code>	set confidence level; default is <code>level(95)</code>
<code>corr</code>	report correlation matrix
<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>noheader</code>	suppress header table from above coefficient table
<code>notable</code>	suppress coefficient table
<code>coeflegend</code>	display legend instead of statistics

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

*depvars* and *indepvars* may contain time-series operators; see [U] 11.4.4 **Time-series varlists**.

`bayes`, `bootstrap`, `by`, `collect`, `jackknife`, `mi estimate`, `rolling`, and `statsby` are allowed; see [U] 11.1.10 **Prefix commands**. For more details, see [BAYES] `bayes: mvreg`.

Weights are not allowed with the `bootstrap` prefix; see [R] `bootstrap`.

`aweight`s are not allowed with the `jackknife` prefix; see [R] `jackknife`.

`aweight`s and `fweight`s are allowed; see [U] 11.1.6 **weight**.

`noheader`, `notable`, and `coeflegend` do not appear in the dialog box.

See [U] 20 **Estimation and postestimation commands** for more capabilities of estimation commands.

## Options

### Model

`noconstant` suppresses the constant term (intercept) in the model.

### Reporting

`level(#)` specifies the confidence level, as a percentage, for confidence intervals. The default is `level(95)` or as set by `set level`; see [U] 20.8 **Specifying the width of confidence intervals**.

`corr` displays the correlation matrix of the residuals between the equations.

`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 options are available with `mvreg` but are not shown in the dialog box:

`noheader` suppresses display of the table reporting  $F$  statistics,  $R^2$ , and root mean squared error above the coefficient table.

`notable` suppresses display of the coefficient table.

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

## Remarks and examples

Multivariate regression differs from multiple regression in that *several* dependent variables are jointly regressed on the same independent variables. Multivariate regression is related to Zellner's seemingly unrelated regression (see [R] [sureg](#)), but because the same set of independent variables is used for each dependent variable, the syntax is simpler, and the calculations are faster.

The individual coefficients and standard errors produced by `mvreg` are identical to those that would be produced by `regress` estimating each equation separately. The difference is that `mvreg`, being a joint estimator, also estimates the between-equation covariances, so you can test coefficients across equations and, in fact, the `test` syntax makes such tests more convenient.

### ► Example 1

Using the automobile data, we fit a multivariate regression for space variables (`headroom`, `trunk`, and `turn`) in terms of a set of other variables, including three performance variables (`displacement`, `gear_ratio`, and `mpg`):

```
. use https://www.stata-press.com/data/r18/auto
(1978 automobile data)
```

```
. mvreg headroom trunk turn = price mpg displ gear_ratio length weight
```

Equation	Obs	Parms	RMSE	"R-sq"	F	P>F
headroom	74	7	.7390205	0.2996	4.777213	0.0004
trunk	74	7	3.052314	0.5326	12.7265	0.0000
turn	74	7	2.132377	0.7844	40.62042	0.0000

  

	Coefficient	Std. err.	t	P> t	[95% conf. interval]	
<b>headroom</b>						
price	-.0000528	.000038	-1.39	0.168	-.0001286 .0000229	
mpg	-.0093774	.0260463	-0.36	0.720	-.061366 .0426112	
displacement	.0031025	.0024999	1.24	0.219	-.0018873 .0080922	
gear_ratio	.2108071	.3539588	0.60	0.553	-.4956976 .9173119	
length	.015886	.012944	1.23	0.224	-.0099504 .0417223	
weight	-.0000868	.0004724	-0.18	0.855	-.0010296 .0008561	
_cons	-.4525117	2.170073	-0.21	0.835	-4.783995 3.878972	
<b>trunk</b>						
price	.0000445	.0001567	0.28	0.778	-.0002684 .0003573	
mpg	-.0220919	.1075767	-0.21	0.838	-.2368159 .1926322	
displacement	.0032118	.0103251	0.31	0.757	-.0173971 .0238207	
gear_ratio	-.2271321	1.461926	-0.16	0.877	-3.145149 2.690885	
length	.170811	.0534615	3.20	0.002	.0641014 .2775206	
weight	-.0015944	.001951	-0.82	0.417	-.0054885 .0022997	
_cons	-13.28253	8.962868	-1.48	0.143	-31.17249 4.607429	
<b>turn</b>						
price	-.0002647	.0001095	-2.42	0.018	-.0004833 -.0000462	
mpg	-.0492948	.0751542	-0.66	0.514	-.1993031 .1007136	
displacement	.0036977	.0072132	0.51	0.610	-.0106999 .0180953	
gear_ratio	-.1048432	1.021316	-0.10	0.919	-2.143399 1.933712	
length	.072128	.0373487	1.93	0.058	-.0024204 .1466764	
weight	.0027059	.001363	1.99	0.051	-.0000145 .0054264	
_cons	20.19157	6.261549	3.22	0.002	7.693467 32.68968	

We should have specified the `corr` option so that we would also see the correlations between the residuals of the equations. We can correct our omission because `mvreg`—like all estimation commands—typed without arguments redisplay results. The `noheader` and `notable` (read “notable”) options suppress redisplaying the output we have already seen:

```
. mvreg, notable noheader corr
Correlation matrix of residuals:
      headroom      trunk      turn
headroom  1.0000
      trunk  0.4986  1.0000
      turn -0.1090 -0.0628  1.0000
Breusch-Pagan test of independence: chi2(3) = 19.566, Pr = 0.0002
```

The Breusch–Pagan test is significant, so the residuals of these three space variables are not independent of each other.

The three performance variables among our independent variables are `mpg`, `displacement`, and `gear_ratio`. We can jointly test the significance of these three variables in all the equations by typing

```
. test mpg displacement gear_ratio
( 1) [headroom]mpg = 0
( 2) [trunk]mpg = 0
( 3) [turn]mpg = 0
( 4) [headroom]displacement = 0
( 5) [trunk]displacement = 0
( 6) [turn]displacement = 0
( 7) [headroom]gear_ratio = 0
( 8) [trunk]gear_ratio = 0
( 9) [turn]gear_ratio = 0
      F( 9, 67) = 0.33
      Prob > F = 0.9622
```

These three variables are not, as a group, significant. We might have suspected this from their individual significance in the individual regressions, but this multivariate test provides an overall assessment with one  $p$ -value.

We can also perform a test for the joint significance of all three equations:

```
. test [headroom]
      (output omitted)
. test [trunk], accum
      (output omitted)
. test [turn], accum
( 1) [headroom]price = 0
( 2) [headroom]mpg = 0
( 3) [headroom]displacement = 0
( 4) [headroom]gear_ratio = 0
( 5) [headroom]length = 0
( 6) [headroom]weight = 0
( 7) [trunk]price = 0
( 8) [trunk]mpg = 0
( 9) [trunk]displacement = 0
(10) [trunk]gear_ratio = 0
(11) [trunk]length = 0
(12) [trunk]weight = 0
(13) [turn]price = 0
(14) [turn]mpg = 0
(15) [turn]displacement = 0
(16) [turn]gear_ratio = 0
(17) [turn]length = 0
(18) [turn]weight = 0
      F( 18,    67) =    19.34
      Prob > F =    0.0000
```

The set of variables as a whole is strongly significant. We might have suspected this, too, from the individual equations. ◀

## □ Technical note

The `mvreg` command provides a good way to deal with multiple comparisons. If we wanted to assess the effect of `length`, we might be dissuaded from interpreting any of its coefficients except that in the `trunk` equation. `[trunk]length`—the coefficient on `length` in the `trunk` equation—has a  $p$ -value of 0.002, but in the other two equations, it has  $p$ -values of only 0.224 and 0.058.

A conservative statistician might argue that there are 18 tests of significance in `mvreg`'s output (not counting those for the intercept), so  $p$ -values more than  $0.05/18 = 0.0028$  should be declared insignificant at the 5% level. A more aggressive but, in our opinion, reasonable approach would be to first note that the three equations are jointly significant, so we are justified in making some interpretation. Then we would work through the individual variables using `test`, possibly using  $0.05/6 = 0.0083$  (6 because there are six independent variables) for the 5% significance level. For instance, examining `length`:

```
. test length
( 1) [headroom]length = 0
( 2) [trunk]length = 0
( 3) [turn]length = 0
      F( 3,    67) =    4.94
      Prob > F =    0.0037
```

The reported significance level of 0.0037 is less than 0.0083, so we will declare this variable significant. `[trunk]length` is certainly significant with its  $p$ -value of 0.002, but what about in the remaining two equations with  $p$ -values 0.224 and 0.058? We perform a joint test:

```

. test [headroom]length [turn]length
( 1) [headroom]length = 0
( 2) [turn]length = 0
      F( 2, 67) = 2.91
      Prob > F = 0.0613

```

At this point, reasonable statisticians could disagree. The 0.06 significance value suggests no interpretation, but these were the two least-significant values out of three, so we would expect the  $p$ -value to be a little high. Perhaps an equivocal statement is warranted: there seems to be an effect, but chance cannot be excluded.



## Stored results

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

### Scalars

<code>e(N)</code>	number of observations
<code>e(k)</code>	number of parameters in each equation
<code>e(k_eq)</code>	number of equations in <code>e(b)</code>
<code>e(df_r)</code>	residual degrees of freedom
<code>e(chi2)</code>	Breusch–Pagan $\chi^2$ (corr only)
<code>e(df_chi2)</code>	degrees of freedom for Breusch–Pagan $\chi^2$ (corr only)
<code>e(rank)</code>	rank of <code>e(V)</code>

### Macros

<code>e(cmd)</code>	<code>mvreg</code>
<code>e(cmdline)</code>	command as typed
<code>e(depvar)</code>	names of dependent variables
<code>e(eqnames)</code>	names of equations
<code>e(wtype)</code>	weight type
<code>e(wexp)</code>	weight expression
<code>e(r2)</code>	$R^2$ for each equation
<code>e(rmse)</code>	RMSE for each equation
<code>e(F)</code>	$F$ statistic for each equation
<code>e(p_F)</code>	$p$ -value for $F$ test for each equation
<code>e(properties)</code>	<code>b V</code>
<code>e(estat_cmd)</code>	program used to implement <code>estat</code>
<code>e(predict)</code>	program used to implement <code>predict</code>
<code>e(marginsok)</code>	predictions allowed by <code>margins</code>
<code>e(marginsnotok)</code>	predictions disallowed by <code>margins</code>
<code>e(marginsdefault)</code>	default <code>predict()</code> specification for <code>margins</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(Sigma)</code>	$\Sigma$ matrix
<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

Given  $q$  equations and  $p$  independent variables (including the constant), the parameter estimates are given by the  $p \times q$  matrix

$$\mathbf{B} = (\mathbf{X}'\mathbf{W}\mathbf{X})^{-1}\mathbf{X}'\mathbf{W}\mathbf{Y}$$

where  $\mathbf{Y}$  is an  $n \times q$  matrix of dependent variables and  $\mathbf{X}$  is a  $n \times p$  matrix of independent variables.  $\mathbf{W}$  is a weighting matrix equal to  $\mathbf{I}$  if no weights are specified. If weights are specified, let  $\mathbf{v}$ :  $1 \times n$  be the specified weights. If `fweight` frequency weights are specified,  $\mathbf{W} = \text{diag}(\mathbf{v})$ . If `aweight` analytic weights are specified,  $\mathbf{W} = \text{diag}\{\mathbf{v}/(\mathbf{1}'\mathbf{v})(\mathbf{1}'\mathbf{1})\}$ , meaning that the weights are normalized to sum to the number of observations.

The residual covariance matrix is

$$\mathbf{R} = \{\mathbf{Y}'\mathbf{W}\mathbf{Y} - \mathbf{B}'(\mathbf{X}'\mathbf{W}\mathbf{X})\mathbf{B}\}/(n - p)$$

The estimated covariance matrix of the estimates is  $\mathbf{R} \otimes (\mathbf{X}'\mathbf{W}\mathbf{X})^{-1}$ . These results are identical to those produced by `sureg` when the same list of independent variables is specified repeatedly; see [R] [sureg](#).

The [Breusch and Pagan \(1980\)](#)  $\chi^2$  statistic—a Lagrange multiplier statistic—is given by

$$\lambda = n \sum_{i=1}^q \sum_{j=1}^{i-1} r_{ij}^2$$

where  $r_{ij}$  is the estimated correlation between the residuals of the equations and  $n$  is the number of observations. It is distributed as  $\chi^2$  with  $q(q - 1)/2$  degrees of freedom.

## References

- Breusch, T. S., and A. R. Pagan. 1980. The Lagrange multiplier test and its applications to model specification in econometrics. *Review of Economic Studies* 47: 239–253. <https://doi.org/10.2307/2297111>.
- Oberfichtner, M., and H. Tauchmann. 2021. Stacked linear regression analysis to facilitate testing of hypotheses across OLS regressions. *Stata Journal* 21: 411–429.

## Also see

- [MV] [mvreg postestimation](#) — Postestimation tools for `mvreg`
- [MV] [manova](#) — Multivariate analysis of variance and covariance
- [BAYES] [bayes: mvreg](#) — Bayesian multivariate regression
- [MI] [Estimation](#) — Estimation commands for use with `mi estimate`
- [R] [nlsur](#) — Estimation of nonlinear systems of equations
- [R] [reg3](#) — Three-stage estimation for systems of simultaneous equations
- [R] [regress](#) — Linear regression
- [R] [regress postestimation](#) — Postestimation tools for `regress`
- [R] [sureg](#) — Zellner’s seemingly unrelated regression
- [SEM] [Intro 5](#) — Tour of models

### [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. 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.

