manova postestimation — Postestimation tools for manova

Postestimation commands | predict | margins
manovatest | test | Remarks and examples
Stored results | Methods and formulas | Also see

Postestimation commands

The following postestimation commands are of special interest after `manova`:

Command	Description
`manovatest`	multivariate tests after `manova`
`screeplot`	plot eigenvalues

The following standard postestimation commands are also available:

Command	Description
`contrast`	contrasts and ANOVA-style joint tests of parameters
`estat summarize`	summary statistics for the estimation sample
`estat vce`	variance–covariance matrix of the estimators (VCE)
`estimates`	cataloging estimation results
`etable`	table of estimation results
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of parameters
`margins`	marginal means, predictive margins, marginal effects, and average marginal effects
`marginsplot`	graph the results from margins (profile plots, interaction plots, etc.)
`nlcom`	point estimates, standard errors, testing, and inference for nonlinear combinations of parameters
`predict`	predictions and their SEs, residuals, etc.
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`pwcompare`	pairwise comparisons of parameters
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

predict

Description for predict

`predict` creates a new variable containing predictions such as fitted values, standard errors, residuals, and differences between the linear predictions.

Menu for predict

Statistics > Postestimation

Syntax for predict

`predict` [*type*] *newvar* [*if*] [*in*] [, *statistic* `equation(`*eqspec*[, *eqspec*]`)`]

statistic	Description
Main	
`xb`	$\mathbf{x}_j\mathbf{b}$, fitted values; the default
`stdp`	standard error of the fitted value
`residuals`	residuals
`difference`	difference between the linear predictions of two equations
`stddp`	standard error of the fitted values for differences

These statistics are available both in and out of sample; type `predict ... if e(sample) ...` if wanted only for the estimation sample.

Options for predict

Main

`xb`, the default, calculates the fitted values—the prediction of $\mathbf{x}_j\mathbf{b}$ for the specified equation.

`stdp` calculates the standard error of the prediction for the specified equation (the standard error of the estimated expected value or mean for the observation's covariate pattern). The standard error of the prediction is also referred to as the standard error of the fitted value.

`residuals` calculates the residuals.

`difference` calculates the difference between the linear predictions of two equations in the system.

`stddp` calculates the standard error of the difference in linear predictions ($\mathbf{x}_{1j}\mathbf{b}-\mathbf{x}_{2j}\mathbf{b}$) between equations 1 and 2.

`equation(`*eqspec*[, *eqspec*]`)` specifies to which equation you are referring. *eqspec* can be an equation number or an equation name. For example, if *eqspec* is a number, then `equation(#1)` would mean that the calculation is to be made for the first equation (that is, for the first dependent variable), `equation(#2)` would mean the second, and so on. If *eqspec* is a name, then `equation(income)` would refer to the equation named `income`, and `equation(hours)` would refer to the equation named `hours`.

If you do not specify `equation()`, results are the same as if you had specified `equation(#1)`.

Specify one equation with the `xb`, `stdp`, or `residuals` statistic.

Specify two equations (for example, `equation(#1, #2)` or `equation(income, hours)`) with the `difference` or `stddp` statistic; these statistics refer to between-equations concepts. `equation()` is required with `difference` and `stddp`.

margins

Description for margins

`margins` estimates margins of responses for linear predictions, fitted values, and differences between the linear predictions.

Menu for margins

Statistics > Postestimation

Syntax for margins

`margins` [*marginlist*] [, *options*]

`margins` [*marginlist*] , `predict(`*statistic* ...`)` [`predict(`*statistic* ...`)` ...] [*options*]

statistic	Description
default	linear predictions for each equation
`xb`	$\mathbf{x}_j\mathbf{b}$, fitted values
`difference`	difference between the linear predictions of two equations
`residuals`	not allowed with `margins`
`stdp`	not allowed with `margins`
`stddp`	not allowed with `margins`

`xb` defaults to the first equation.

Statistics not allowed with `margins` are functions of stochastic quantities other than `e(b)`.

For the full syntax, see [R] **margins**.

manovatest

Description of manovatest

`manovatest` provides multivariate tests involving *term*s or linear combinations of the underlying design matrix from the most recently fit `manova`. The four multivariate test statistics are Wilks's lambda, Pillai's trace, Lawley–Hotelling trace, and Roy's largest root. The format of the output is similar to that shown by `manova`; see [MV] **manova**.

Menu for manovatest

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

Syntax for manovatest

`manovatest` *term* [*term* ...] [`/` *term* [*term* ...]] [`,` `ytransform(`*matname*`)`]

`manovatest , test(`*matname*`)` [`ytransform(`*matname*`)`]

`manovatest , showorder`

where *term* is a term from the *termlist* in the previously run `manova`.

`collect` is allowed with `manovatest`; see **[U] 11.1.10 Prefix commands**.

Options for manovatest

`ytransform(`*matname*`)` specifies a matrix for transforming the y variables (the *depvars* from `manova`) as part of the test. The multivariate tests are based on $(\mathbf{AEA}')^{-1}(\mathbf{AHA}')$. By default, $\mathbf{A}$ is the identity matrix. `ytransform()` is how you specify an $\mathbf{A}$ matrix to be used in the multivariate tests. Specifying `ytransform()` provides the same results as first transforming the y variables with $\mathbf{YA}'$, where $\mathbf{Y}$ is the matrix formed by binding the y variables by column and $\mathbf{A}$ is the matrix stored in *matname*; then performing `manova` on the transformed y's; and finally running `manovatest` without `ytransform()`.

The number of columns of *matname* must equal the number of variables in the *depvars* from `manova`. The number of rows must be less than or equal to the number of variables in the *depvars* from `manova`. *matname* should have columns in the same order as the *depvars* from `manova`. The column and row names of *matname* are ignored.

When `ytransform()` is specified, a listing of the transformations is presented before the table containing the multivariate tests. You should examine this table to verify that you have applied the transformation you desired.

`test(`*matname*`)` is required with the second syntax of `manovatest`. The rows of *matname* specify linear combinations of the underlying design matrix of the MANOVA that are to be jointly tested. The columns correspond to the underlying design matrix (including the constant if it has not been suppressed). The column and row names of *matname* are ignored.

A listing of the constraints imposed by the `test()` option is presented before the table containing the multivariate tests. You should examine this table to verify that you have applied the linear combinations you desired. Typing `manovatest, showorder` allows you to examine the ordering of the columns for the design matrix from the MANOVA.

`showorder` causes `manovatest` to list the definition of each column in the design matrix. `showorder` is not allowed with any other option or when *term*s are specified.

test

Description for test

In addition to the standard syntax of `test` (see [R] **test**), `test` after `manova` has two additionally allowed syntaxes; see below. `test` performs Wald tests of expressions involving the coefficients of the underlying regression model. Simple and composite linear hypotheses are possible.

Menu for test

Statistics > Multivariate analysis > MANOVA, multivariate regression, and related > Wald test after MANOVA

Syntax for test

In addition to the standard syntax of `test` (see [R] **test**), `test` after `manova` also allows the following.

`test , test(`*matname*`) [ mtest[(`*opt*`)] matvlc(`*matname*`) ]`	syntax A
`test , showorder`	syntax B

syntax A — test expression involving the coefficients of the underlying multivariate regression model; you provide information as a matrix

syntax B — show underlying order of design matrix, which is useful when constructing the *matname* argument of the `test()` option

Options for test

Main

`test(`*matname*`)` is required with syntax A of `test`. The rows of *matname* specify linear combinations of the underlying design matrix of the MANOVA that are to be jointly tested. The columns correspond to the underlying design matrix (including the constant if it has not been suppressed). The column and row names of *matname* are ignored.

A listing of the constraints imposed by the `test()` option is presented before the table containing the tests. You should examine this table to verify that you have applied the linear combinations you desired. Typing `test, showorder` allows you to examine the ordering of the columns for the design matrix from the MANOVA.

matname should have as many columns as the number of dependent variables times the number of columns in the basic design matrix. The design matrix is repeated for each dependent variable.

`showorder` causes `test` to list the definition of each column in the design matrix. `showorder` is not allowed with any other option.

Options

`mtest[(`*opt*`)]` specifies that tests be performed for each condition separately. *opt* specifies the method for adjusting p-values for multiple testing. Valid values for *opt* are

`bonferroni`	Bonferroni's method
`holm`	Holm's method
`sidak`	Šidák's method
`noadjust`	no adjustment is to be made

Specifying `mtest` without an argument is equivalent to specifying `mtest(noadjust)`.

The following option is available with `test` after `manova` but is not shown in the dialog box:

`matvlc(`*matname*`)`, a programmer's option, saves the variance–covariance matrix of the linear combinations involved in the suite of tests. For the test of H_0: $\mathbf{Lb} = \mathbf{c}$, what is returned in *matname* is $\mathbf{LVL}'$, where $\mathbf{V}$ is the estimated variance–covariance matrix of $\mathbf{b}$.

Remarks and examples

Several postestimation tools are available after `manova`. We demonstrate these tools by extending examples 1, 2, 4, and 8 of [MV] **manova**.

▷ Example 1: test and lincom

Example 1 of [MV] **manova** presented a balanced one-way MANOVA on the rootstock data.

```
. use https://www.stata-press.com/data/r19/rootstock
(Table 6.2. Rootstock data, Rencher and Christensen (2012))
. manova y1 y2 y3 y4 = rootstock
  (output omitted)
```

`test` provides Wald tests on expressions involving the underlying coefficients of the model, and `lincom` provides linear combinations along with standard errors and confidence intervals.

```
. test [y3]3.rootstock = ([y3]1.rootstock + [y3]2.rootstock)/2
 ( 1)  - .5*[y3]1b.rootstock - .5*[y3]2.rootstock + [y3]3.rootstock = 0
       F(  1,    42) =    5.62
            Prob > F =    0.0224
. lincom [y3]4.rootstock - [y1]4.rootstock
 ( 1)  - [y1]4.rootstock + [y3]4.rootstock = 0
```

	Coefficient	Std. err.	t	P>|t|	[95% conf.	interval]
(1)	.2075001	.1443917	1.44	0.158	-.0838941	.4988943

If the equation portion of the expression is omitted, the first equation (first dependent variable) is assumed.

◁

The manovatest postestimation command provides multivariate tests of *term*s or linear combinations of the underlying design matrix from the most recent MANOVA model.

▷ Example 2: manovatest

In example 2 of [MV] **manova**, a one-way MANOVA on the metabolic dataset was shown.

```
. use https://www.stata-press.com/data/r19/metabolic
(Table 4.5. Metabolic comparisons of rabbits, Rencher (1998))
. manova y1 y2 = group
  (output omitted)
```

manovatest can test *term*s from the preceding manova. Here we test the group term from our one-way MANOVA:

```
. manovatest group
                W = Wilks' lambda      L = Lawley-Hotelling trace
                P = Pillai's trace     R = Roy's largest root
------------------------------------------------------------------------------
      Source | Statistic        df     F(df1,     df2) =    F    Prob>F
-------------+----------------------------------------------------------------
       group | W   0.1596        3       6.0      32.0      8.02 0.0000 e
             | P   1.2004                6.0      34.0      8.51 0.0000 a
             | L   3.0096                6.0      30.0      7.52 0.0001 a
             | R   1.5986                3.0      17.0      9.06 0.0008 u
-------------+----------------------------------------------------------------
    Residual |                  17
------------------------------------------------------------------------------
                e = exact, a = approximate, u = upper bound on F
```

Using manovatest to test model *term*s is not interesting here. It merely repeats information already presented by manova. Later we will see useful applications of *term* testing via manovatest.

manovatest can also be used to test linear combinations of the underlying design matrix of the MANOVA model. Whereas the MANOVA indicates that there are differences in the groups, it does not indicate the nature of those differences. Rencher discusses three linear contrasts of interest for this example: group one (the control) versus the rest, group four versus groups two and three, and group two versus group three. The test() option of manovatest allows us to test these hypotheses.

Because we did not use the noconstant option with our manova, the underlying parameterization of the design matrix has the last column corresponding to the constant in the model, whereas the first four columns correspond to the four groups of rabbits. The showorder option of manovatest illustrates this point. The tests on the three contrasts of interest follow.

```
. manovatest, showorder
 Order of columns in the design matrix
      1: (group==1)
      2: (group==2)
      3: (group==3)
      4: (group==4)
      5: _cons
. matrix c1 = (3,-1,-1,-1,0)
```

```
. manovatest, test(c1)

 Test constraint
 (1)    3*1.group - 2.group - 3.group - 4.group = 0

                       W = Wilks' lambda      L = Lawley-Hotelling trace
                       P = Pillai's trace     R = Roy's largest root

             Source | Statistic        df    F(df1,     df2) =   F   Prob>F
         ----------+-------------------------------------------------------
         manovatest |W   0.4063         1      2.0     16.0    11.69 0.0007 e
                    |P   0.5937                2.0     16.0    11.69 0.0007 e
                    |L   1.4615                2.0     16.0    11.69 0.0007 e
                    |R   1.4615                2.0     16.0    11.69 0.0007 e
                    |-------------------------------------------------------
           Residual |                  17
         ----------+-------------------------------------------------------

                       e = exact, a = approximate, u = upper bound on F

. matrix c2 = (0,-1,-1,2,0)

. manovatest, test(c2)

 Test constraint
 (1)    - 2.group - 3.group + 2*4.group = 0

                       W = Wilks' lambda      L = Lawley-Hotelling trace
                       P = Pillai's trace     R = Roy's largest root

             Source | Statistic        df    F(df1,     df2) =   F   Prob>F
         ----------+-------------------------------------------------------
         manovatest |W   0.9567         1      2.0     16.0     0.36 0.7018 e
                    |P   0.0433                2.0     16.0     0.36 0.7018 e
                    |L   0.0453                2.0     16.0     0.36 0.7018 e
                    |R   0.0453                2.0     16.0     0.36 0.7018 e
                    |-------------------------------------------------------
           Residual |                  17
         ----------+-------------------------------------------------------

                       e = exact, a = approximate, u = upper bound on F

. matrix c3 = (0,1,-1,0,0)

. manovatest, test(c3)

 Test constraint
 (1)    2.group - 3.group = 0

                       W = Wilks' lambda      L = Lawley-Hotelling trace
                       P = Pillai's trace     R = Roy's largest root

             Source | Statistic        df    F(df1,     df2) =   F   Prob>F
         ----------+-------------------------------------------------------
         manovatest |W   0.4161         1      2.0     16.0    11.23 0.0009 e
                    |P   0.5839                2.0     16.0    11.23 0.0009 e
                    |L   1.4033                2.0     16.0    11.23 0.0009 e
                    |R   1.4033                2.0     16.0    11.23 0.0009 e
                    |-------------------------------------------------------
           Residual |                  17
         ----------+-------------------------------------------------------

                       e = exact, a = approximate, u = upper bound on F
```

Because there is only 1 degree of freedom for each of the hypotheses, the F tests are exact (and identical for the four multivariate methods). The first test indicates that the mean vector for the control group is significantly different from the mean vectors for the other three groups. The second test, with a p-value of 0.7018, fails to reject the null hypothesis that group four equals groups two and three. The third test, with a p-value of 0.0009, indicates differences between the mean vectors of groups two and three.

Rencher also tests using weighted orthogonal contrasts. `manovatest` can do these tests as well.

```
. matrix c1w = (14,-7,-5,-2,0)
. manovatest, test(c1w)
 Test constraint
 (1)    14*1.group - 7*2.group - 5*3.group - 2*4.group = 0
                      W = Wilks' lambda      L = Lawley-Hotelling trace
                      P = Pillai's trace     R = Roy's largest root
      Source | Statistic        df    F(df1,     df2) =   F   Prob>F
-------------+-----------------------------------------------------
  manovatest | W   0.3866         1      2.0     16.0    12.70 0.0005 e
             | P   0.6134                2.0     16.0    12.70 0.0005 e
             | L   1.5869                2.0     16.0    12.70 0.0005 e
             | R   1.5869                2.0     16.0    12.70 0.0005 e
-------------+-----------------------------------------------------
    Residual |                   17
-------------+-----------------------------------------------------
                      e = exact, a = approximate, u = upper bound on F
. matrix c2w = (0,-7,-5,12,0)
. manovatest, test(c2w)
 Test constraint
 (1)    -7*2.group - 5*3.group + 12*4.group = 0
                      W = Wilks' lambda      L = Lawley-Hotelling trace
                      P = Pillai's trace     R = Roy's largest root
      Source | Statistic        df    F(df1,     df2) =   F   Prob>F
-------------+-----------------------------------------------------
  manovatest | W   0.9810         1      2.0     16.0     0.15 0.8580 e
             | P   0.0190                2.0     16.0     0.15 0.8580 e
             | L   0.0193                2.0     16.0     0.15 0.8580 e
             | R   0.0193                2.0     16.0     0.15 0.8580 e
-------------+-----------------------------------------------------
    Residual |                   17
-------------+-----------------------------------------------------
                      e = exact, a = approximate, u = upper bound on F
```

These two weighted contrasts do not lead to different conclusions compared with their unweighted counterparts.

◁

❑ Technical note

`manovatest, test(`*matname*`)` displays the linear combination (labeled "Test constraint") indicated by *matname*. You should examine this listing to make sure that the matrix you specify in `test()` provides the test you want.

❑

The `margins` postestimation command provides, among other things, tables of predicted means and confidence intervals that are based on the most recently fit model.

▷ Example 3: Margins, contrasts, and predictions

Example 4 of [MV] **manova** presented a two-way MANOVA model on the jaw data.

```
. use https://www.stata-press.com/data/r19/jaw
(Table 4.6. Two-way unbalanced data for fractures of the jaw, Rencher (1998))
. manova y1 y2 y3 = gender fracture gender#fracture
  (output omitted)
```

The interaction term, gender#fracture, was significant. margins may be used to examine the interaction; see [R] **margins**.

```
. margins gender#fracture, predict(equation(y1))
Adjusted predictions                                    Number of obs = 27
Expression: Linear prediction, predict(equation(y1))
```

	Margin	Delta-method std. err.	t	P>|t|	[95% conf.	interval]
gender# fracture						
Male # One compou..	39.5	4.171386	9.47	0.000	30.82513	48.17487
Male # Two compou..	26.875	3.612526	7.44	0.000	19.36234	34.38766
Male # One simple..	45.16667	4.171386	10.83	0.000	36.49179	53.84154
Female # One compou..	22	10.21777	2.15	0.043	.7509887	43.24901
Female # Two compou..	30.75	5.108884	6.02	0.000	20.12549	41.37451
Female # One simple..	36.5	7.225053	5.05	0.000	21.47468	51.52532

```
. margins gender#fracture, predict(equation(y2))
Adjusted predictions                                    Number of obs = 27
Expression: Linear prediction, predict(equation(y2))
```

	Margin	Delta-method std. err.	t	P>|t|	[95% conf.	interval]
gender# fracture						
Male # One compou..	35.5	2.150966	16.50	0.000	31.02682	39.97318
Male # Two compou..	32.375	1.862791	17.38	0.000	28.50111	36.24889
Male # One simple..	36.16667	2.150966	16.81	0.000	31.69349	40.63984
Female # One compou..	56	5.268768	10.63	0.000	45.043	66.957
Female # Two compou..	33.25	2.634384	12.62	0.000	27.7715	38.7285
Female # One simple..	33	3.725582	8.86	0.000	25.25223	40.74777

```
. margins gender#fracture, predict(equation(y3))
Adjusted predictions                                    Number of obs = 27
Expression: Linear prediction, predict(equation(y3))
```

	Margin	Delta-method std. err.	t	P>|t|	[95% conf.	interval]
gender# fracture						
Male # One compou..	61.16667	2.038648	30.00	0.000	56.92707	65.40627
Male # Two compou..	62.25	1.765521	35.26	0.000	58.5784	65.9216
Male # One simple..	58.16667	2.038648	28.53	0.000	53.92707	62.40627
Female # One compou..	43	4.993647	8.61	0.000	32.61514	53.38486
Female # Two compou..	64	2.496823	25.63	0.000	58.80757	69.19243
Female # One simple..	63.5	3.531041	17.98	0.000	56.1568	70.8432

The first `margins` table shows the predicted mean (marginal mean), standard error, z statistic, p-value, and confidence interval of `y1` (age of patient) for each combination of `fracture` and `gender`. The second and third `margins` tables provide this information for `y2` (blood lymphocytes) and `y3` (blood polymorphonuclears). These three tables of predictions are the same as those you would obtain from `margins` after running `anova` for each of the three dependent variables separately.

The predicted `y2` value is larger than the predicted `y3` value for females with one compound fracture. For the other five combinations of `gender` and `fracture`, the relationship is reversed. There is only 1 observation for the combination of female and one compound fracture.

There are nine possible contrasts if we contrast women with men for every fracture type and every dependent variable. We will use `contrast` to estimate all nine contrasts and apply Scheffé's adjustment for multiple comparisons.

```
. contrast gender@fracture#_eqns, mcompare(scheffe)
Contrasts of marginal linear predictions
Margins: asbalanced
```

	df	F	P>F	Scheffe P>F
gender@fracture#_eqns				
One compound fracture 1	1	2.51	0.1278	0.9733
One compound fracture 2	1	12.98	0.0017	0.2333
One compound fracture 3	1	11.34	0.0029	0.3137
Two compound fractures 1	1	0.38	0.5424	1.0000
Two compound fractures 2	1	0.07	0.7889	1.0000
Two compound fractures 3	1	0.33	0.5732	1.0000
One simple fracture 1	1	1.08	0.3107	0.9987
One simple fracture 2	1	0.54	0.4698	0.9999
One simple fracture 3	1	1.71	0.2050	0.9929
Joint	9	2.57	0.0361	
Denominator	21			

```
Note: Scheffe-adjusted p-values are reported for tests on individual
      contrasts only.
```

	Number of comparisons
gender@fracture#_eqns	9

	Contrast	Std. err.	Scheffe [95% conf.	interval]
gender@fracture#_eqns				
(Female vs base)				
One compound fracture 1	-17.5	11.03645	-68.42869	33.42869
(Female vs base)				
One compound fracture 2	20.5	5.69092	-5.76126	46.76126
(Female vs base)				
One compound fracture 3	-18.16667	5.393755	-43.05663	6.723297
(Female vs base)				
Two compound fractures 1	3.875	6.257079	-24.99885	32.74885
(Female vs base)				
Two compound fractures 2	.875	3.226449	-14.01373	15.76373
(Female vs base)				
Two compound fractures 3	1.75	3.057972	-12.36128	15.86128
(Female vs base)				
One simple fracture 1	-8.666667	8.342772	-47.16513	29.8318
(Female vs base)				
One simple fracture 2	-3.166667	4.301931	-23.01831	16.68498
(Female vs base)				
One simple fracture 3	5.333333	4.077296	-13.48171	24.14838

Women do not differ significantly from men in any of the nine comparisons.

Let's examine the residuals with the `predict` command:

```
. predict y1res, residual equation(y1)
. predict y2res, residual equation(y2)
. predict y3res, residual equation(y3)
. list gender fracture y1res y2res y3res
```

	gender	fracture	y1res	y2res	y3res
1.	Male	One compound fracture	2.5	-.5	-.1666667
2.	Male	One compound fracture	2.5	7.5	-6.166667
3.	Male	One compound fracture	8.5	-.5	2.833333
4.	Male	One compound fracture	-4.5	-2.5	3.833333
5.	Male	One compound fracture	-14.5	-4.5	2.833333
6.	Male	One compound fracture	5.5	.5	-3.166667
7.	Male	Two compound fractures	-3.875	-5.375	1.75
8.	Male	Two compound fractures	-4.875	-.375	1.75
9.	Male	Two compound fractures	-1.875	-2.375	1.75
10.	Male	Two compound fractures	1.125	6.625	-6.25
11.	Male	Two compound fractures	-2.875	-1.375	6.75
12.	Male	Two compound fractures	25.125	-4.375	-2.25
13.	Male	Two compound fractures	-9.875	-2.375	1.75
14.	Male	Two compound fractures	-2.875	9.625	-5.25
15.	Male	One simple fracture	-13.16667	.8333333	-4.166667
16.	Male	One simple fracture	6.833333	-2.166667	3.833333
17.	Male	One simple fracture	7.833333	8.833333	-7.166667
18.	Male	One simple fracture	3.833333	-1.166667	1.833333
19.	Male	One simple fracture	9.833333	-4.166667	1.833333
20.	Male	One simple fracture	-15.16667	-2.166667	3.833333
21.	Female	One compound fracture	0	0	-2.84e-14
22.	Female	Two compound fractures	-8.75	-4.25	4
23.	Female	Two compound fractures	7.25	-8.25	9
24.	Female	Two compound fractures	-9.75	3.75	-5
25.	Female	Two compound fractures	11.25	8.75	-8
26.	Female	One simple fracture	6.5	-3	3.5
27.	Female	One simple fracture	-6.5	3	-3.5

The single observation for a female with one compound fracture has residuals that are within roundoff of zero. With only 1 observation for that cell of the design, this MANOVA model is forced to fit to that point. The largest residual (in absolute value) appears for observation 12, which has an age 25.125 higher than the model prediction for a male with two compound fractures.

◁

▷ Example 4: manovatest

Example 8 of [MV] **manova** presents a nested MANOVA on the `videotrainer` data.

```
. use https://www.stata-press.com/data/r19/videotrainer, clear
(Video training)
. manova primary extra = video / store|video / associate|store|video /,
> dropemptycells
```

(*output omitted*)

The MANOVA indicated that `store` was not significant.

You decide to follow the rule of thumb that says to pool terms whose p-values are larger than 0.25. Wilks's lambda reports a p-value of 0.5775 for the test of `store|video` (see example 8 of [MV] **manova**). You decide to pool the `store` and `associate` terms in the MANOVA to gain power for the test of `video`. The forward-slash notation of `manova` is also allowed with `manovatest` to indicate nonresidual error terms. Here is the multivariate test of `video` using the pooled `store` and `associate` terms and then the multivariate test of the pooled term:

```
. manovatest video / store|video associate|store|video
                    W = Wilks' lambda      L = Lawley-Hotelling trace
                    P = Pillai's trace     R = Roy's largest root
------------------------------------------------------------------------------
      Source | Statistic        df    F(df1,     df2) =   F   Prob>F
-------------+----------------------------------------------------------------
       video | W   0.4079        1       2.0      9.0     6.53 0.0177 e
             | P   0.5921                2.0      9.0     6.53 0.0177 e
             | L   1.4516                2.0      9.0     6.53 0.0177 e
             | R   1.4516                2.0      9.0     6.53 0.0177 e
-------------+----------------------------------------------------------------
 store|video  |                  10
associate|   |
store|video  |
------------------------------------------------------------------------------
                    e = exact, a = approximate, u = upper bound on F
. manovatest store|video associate|store|video
                    W = Wilks' lambda      L = Lawley-Hotelling trace
                    P = Pillai's trace     R = Roy's largest root
------------------------------------------------------------------------------
      Source | Statistic        df    F(df1,     df2) =   F   Prob>F
-------------+----------------------------------------------------------------
 store|video  | W   0.3925       10      20.0     58.0     1.73 0.0546 e
associate|   | P   0.7160                20.0     60.0     1.67 0.0647 a
store|video  | L   1.2711                20.0     56.0     1.78 0.0469 a
             | R   0.9924                10.0     30.0     2.98 0.0100 u
-------------+----------------------------------------------------------------
    Residual |                  30
------------------------------------------------------------------------------
                    e = exact, a = approximate, u = upper bound on F
```

Pooling `store` with `associate` helps increase the power for the test of `video`.

You can show the univariate analysis for one of your dependent variables by using the `ytransform()` option of `manovatest`:

```
. mat primonly = (1,0)
. manovatest video / store|video associate|store|video, ytransform(primonly)
Transformation of the dependent variables
(1)    primary
                    W = Wilks' lambda      L = Lawley-Hotelling trace
                    P = Pillai's trace     R = Roy's largest root
          Source | Statistic        df    F(df1,     df2) =   F   Prob>F
-----------------+-------------------------------------------------------
           video | W   0.8449         1      1.0     10.0     1.84 0.2053 e
                 | P   0.1551                1.0     10.0     1.84 0.2053 e
                 | L   0.1835                1.0     10.0     1.84 0.2053 e
                 | R   0.1835                1.0     10.0     1.84 0.2053 e
-----------------+-------------------------------------------------------
     store|video |                   10
associate|store| |
           video |
-----------------+-------------------------------------------------------
                    e = exact, a = approximate, u = upper bound on F
. manovatest store|video associate|store|video, ytransform(primonly)
Transformation of the dependent variables
(1)    primary
                    W = Wilks' lambda      L = Lawley-Hotelling trace
                    P = Pillai's trace     R = Roy's largest root
          Source | Statistic        df    F(df1,     df2) =   F   Prob>F
-----------------+-------------------------------------------------------
     store|video | W   0.6119        10     10.0     30.0     1.90 0.0846 e
associate|store| | P   0.3881               10.0     30.0     1.90 0.0846 e
           video | L   0.6344               10.0     30.0     1.90 0.0846 e
                 | R   0.6344               10.0     30.0     1.90 0.0846 e
-----------------+-------------------------------------------------------
        Residual |                   30
-----------------+-------------------------------------------------------
                    e = exact, a = approximate, u = upper bound on F
```

See the second `manova` run from example 8 of [MV] **manova** for an alternate way of pooling the terms by refitting the MANOVA model.

◁

See examples 6, 10, 11, 12, and 13 of [MV] **manova** for more examples of `manovatest`, including examples involving both the `test()` and the `ytransform()` options.

Stored results

`manovatest` stores the following in `r()`:

Scalars
- `r(df)` hypothesis degrees of freedom
- `r(df_r)` residual degrees of freedom

Matrices
- `r(H)` hypothesis SSCP matrix
- `r(E)` residual-error SSCP matrix
- `r(stat)` multivariate statistics
- `r(eigvals)` eigenvalues of $\mathtt{E}^{-1}\mathtt{H}$
- `r(aux)` `s`, `m`, and `n` values

`test` after `manova` stores the following in `r()`:

Scalars
- `r(p)` two-sided p-value
- `r(F)` F statistic
- `r(df)` hypothesis degrees of freedom
- `r(df_r)` residual degrees of freedom
- `r(drop)` `0` if no constraints dropped, `1` otherwise
- `r(dropped_#)` index of #th constraint dropped

Macros
- `r(mtmethod)` method of adjustment for multiple testing

Matrices
- `r(mtest)` multiple test results

Methods and formulas

See [MV] **manova** for methods and formulas for the multivariate tests performed by `manovatest`.

Also see

[MV] **manova** — Multivariate analysis of variance and covariance

[MV] **screeplot** — Scree plot of eigenvalues

[U] 20 Estimation and postestimation commands

Stata, Stata Press, Mata, NetCourse, and NetCourseNow are registered trademarks of StataCorp LLC. Stata and Stata Press are registered trademarks with the World Intellectual Property Organization of the United Nations. StataNow is a trademark 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.