Stata Library

How can I form various tests comparing the different levels of a categorical variable after anova or regress?

This page was adapted from a FAQ at the Stata Corp. FAQ page. We thank Stata for their permission to adapt and distribute this page via our web site.

Introduction

Often researchers want to test for differences between levels of a factor (categorical variable) or factors after running an **anova** or **regress** command. For instance, with one factor the questions might be

- Is level one different from level two?
- Is level one different from level three?
- …

There are often other kinds of tests between levels of a factor that are also of interest. For instance, the questions might be of the form:

- Is level one different from the average of levels two through four?
- Is level two different from the average of levels three and four?
- Is level three different from level four?

There are many other interesting questions like these that are possible to ask after an estimation command involving a categorical variable (factor).

The **test** command is one tool to use in answering these questions. There are several variations of the syntax for **test** depending on if you wish to test coefficients, expressions, terms (after **anova**), or to test several coefficients at the same time. The form of the **test** command that we will use is

```
test [exp = exp] [, accumulate notest ]
```

Details can be found in the Reference Manual ([R] **test** and in the case of **anova** the sections in [R] **anova** that discuss testing).

One categorical factor

The way in which you parameterize your model makes a difference in what you do to perform your tests. We will examine a couple of ways of parameterizing a simple one-way ANOVA model. We will use the following three approaches:

- **regress** with the **noconstant** option
- **regress** leaving the constant in the model
- **anova**

With each of these approaches we will show how to use the **test** command to obtain tests of interest. In particular we will

1. test level one against level two
2. test the average of level one and two against level three
3. test that five times level one plus four times level two minus three times level three equals two times level four (a strange linear combination just to show that it can be done)

Here is a 20 observation dataset with two variables, the outcome **y** and a categorical variable **x** with four levels.

```
table x, c(mean y)
```

```
----------+-----------
        x |    mean(y)
----------+-----------
        1 |        4.4
        2 |        3.4
        3 |        5.6
        4 |        6.4
----------+-----------
```

For **regress** we need to create the indicator (sometimes called "dummy") variables corresponding to our variable **x**. We can use **xi** to create the indicator variables (**xi** automatically omits one level). Instead, we will use the **generate()** option of **tabulate** to produce the indicator variables since it does not automatically omit a level.

```
quietly tabulate x, gen(cat)
list, noobs
```

```
      x          y      cat1      cat2      cat3      cat4
      1          7         1         0         0         0
      1          5         1         0         0         0
      1          3         1         0         0         0
      1          4         1         0         0         0
      1          3         1         0         0         0
      2          5         0         1         0         0
      2          3         0         1         0         0
      2          5         0         1         0         0
      2          3         0         1         0         0
      2          1         0         1         0         0
      3          6         0         0         1         0
      3          8         0         0         1         0
      3          6         0         0         1         0
      3          4         0         0         1         0
      3          4         0         0         1         0
      4          5         0         0         0         1
      4          8         0         0         0         1
      4          6         0         0         0         1
      4          8         0         0         0         1
      4          5         0         0         0         1
```

Regression excluding the intercept

Here is the regression excluding the intercept. Note that all four levels of **x** are included in the regression model (possible because we omit the constant).

```
regress y cat*, noconstant
```

```
  Source |       SS       df       MS                  Number of obs =      20
---------+------------------------------               F(  4,    16) =   48.24
   Model |      516.20     4      129.05                Prob > F      =  0.0000
Residual |       42.80    16       2.675               R-squared     =  0.9234
---------+------------------------------               Adj R-squared =  0.9043
   Total |      559.00    20       27.95               Root MSE      =  1.6355

------------------------------------------------------------------------------
       y |      Coef.   Std. Err.       t     P>|t|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
    cat1 |        4.4   .7314369      6.016   0.000       2.849423    5.950577
    cat2 |        3.4   .7314369      4.648   0.000       1.849423    4.950577
    cat3 |        5.6   .7314369      7.656   0.000       4.049423    7.150577
    cat4 |        6.4   .7314369      8.750   0.000       4.849423    7.950577
------------------------------------------------------------------------------
```

Notice how the coefficients agree with the means reported by **table**. These coefficients are easily interpreted and easily tested. Here are the three tests after this regression:

1. test level one against level two

```
test cat1 = cat2

( 1)  cat1 - cat2 = 0.0

      F(  1,    16) =    0.93
           Prob > F =    0.3481
```

2. test the average of level one and two against level three

```
test 0.5*cat1 + 0.5*cat2 = cat3

( 1)  .5 cat1 + .5 cat2 - cat3 = 0.0

      F(  1,    16) =    3.60
           Prob > F =    0.0759
```

3. test that five times level one plus four times level two minus three times level three equals two times level four

```
test 5*cat1 + 4*cat2 - 3*cat3 = 2*cat4

( 1)  5.0 cat1 + 4.0 cat2 - 3.0 cat3 - 2.0 cat4 = 0.0

      F(  1,    16) =    1.25
           Prob > F =    0.2808
```

Any of a number of other strange and/or wonderful tests could be performed. If you wish to test a nonlinear expression you will want to look at **testnl** (see [R] **testnl**). If you want to jointly test two or more of these single degree-of-freedom tests you can use the **accumulate** option of **test**. Another useful command is **lincom** (see [R] **lincom**). It can also be used to test many of the same hypotheses as the **test** command and has the benefit of providing not only the test result but the estimate of the linear combination and the standard error of the estimate along with confidence intervals. However, **lincom** will not allow additive constants and can not be used after **anova**. Just as an example, here is that third test done using **lincom**.

```
lincom 5*cat1 + 4*cat2 - 3*cat3 - 2*cat4

 ( 1)  5.0 cat1 + 4.0 cat2 - 3.0 cat3 - 2.0 cat4 = 0.0

------------------------------------------------------------------------------
       y |      Coef.   Std. Err.       t     P>|t|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
     (1) |          6   5.374942      1.116   0.281      -5.394368    17.39437
------------------------------------------------------------------------------
```

Regression with the intercept

Now, what about this same problem, but after a regression that includes the constant? If you run

```
regress y cat*
```

one of the category variables will be dropped due to collinearity. We can force **regress** to drop the first category (for example) by leaving that indicator variable out of the model.

```
regress y cat2-cat4

  Source |       SS       df       MS                Number of obs =      20
---------+------------------------------             F(  3,    16) =    3.26
   Model |       26.15     3  8.71666667             Prob > F      =  0.0492
Residual |       42.80    16       2.675             R-squared     =  0.3793
---------+------------------------------             Adj R-squared =  0.2629
   Total |       68.95    19  3.62894737             Root MSE      =  1.6355

------------------------------------------------------------------------------
       y |      Coef.   Std. Err.       t     P>|t|       [95% Conf. Interval]
---------+--------------------------------------------------------------------
    cat2 |         -1   1.034408     -0.967   0.348      -3.192847    1.192847
    cat3 |        1.2   1.034408      1.160   0.263      -.9928471    3.392847
    cat4 |          2   1.034408      1.933   0.071      -.1928471    4.192847
   _cons |        4.4   .7314369      6.016   0.000       2.849423    5.950577
------------------------------------------------------------------------------
```

Notice that if you compare the coefficients from this regression to the output from the regression without a constant (and to the table showing the mean for each category), you see that the mean of the dropped category (one) corresponds to the coefficient for the constant and that if you add the coefficient for the constant to the other category coefficients you get the mean for those categories. In other words, the coefficients for the included levels are relative to the dropped level.

This provides the clue on how to obtain our three example tests of interest when the constant is in the model. It is a matter of doing some simple algebra to take a test from the first regression model and produce an equivalent test in this regression model. For reference here is the correspondence between the two regression models:

No constant	**With constant**
cat1	_cons
cat2	_cons + cat2
cat3	_cons + cat3
cat4	_cons + cat4

After our first regression (without a constant) the first test was **test cat1 = cat2**, After this latest regression (with a constant) the same test is **test _cons = _cons + cat2**. This test can be simplified to **test cat2**. The equivalent second and third tests can be determined in a similar fashion. Here are the three tests after **regress** with the constant included:

1. test level one against level two

```
test cat2

( 1)  cat2 = 0.0

      F(  1,    16) =    0.93
           Prob > F =    0.3481
```

2. test the average of level one and two against level three

```
test 0.5*cat2 = cat3

( 1)  .5 cat2 - cat3 = 0.0

      F(  1,    16) =    3.60
           Prob > F =    0.0759
```

3. test that five times level one plus four times level two minus three times level three equals two times level four

```
test 4*_cons + 4*cat2 - 3*cat3 = 2*cat4

( 1)  4.0 cat2 - 3.0 cat3 - 2.0 cat4 + 4.0 _cons = 0.0

      F(  1,    16) =    1.25
           Prob > F =    0.2808
```

When you have the constant in the model (and in more complicated designs) it is important to understand how to interpret the coefficients from the regression model so that you can form the correct tests. The formation of the **test** above is not very intuitive, but becomes clearer after doing some algebra starting with your understanding of the meaning of each coefficient and how they relate to the quantities you wish to test.

ANOVA

The **anova** command is a natural choice for analyzing this same data.

```
anova  y  x

                          Number of obs =      20     R-squared     =  0.3793
                          Root MSE      = 1.63554     Adj R-squared =  0.2629

                 Source |  Partial SS    df       MS           F     Prob > F
            ------------+----------------------------------------------------
                  Model |       26.15     3  8.71666667       3.26     0.0492
                        |
                      x |       26.15     3  8.71666667       3.26     0.0492
                        |
                Residual |       42.80    16       2.675
            ------------+----------------------------------------------------
                  Total |       68.95    19  3.62894737
```

We will examine obtaining individual degree-of-freedom tests in just a moment. First we will take a look at the underlying regression for this **anova**. You can get it in several ways. You can simply type **regress** after an **anova** or you can type **anova, regress** (or you could have specified the **regress** option when originally running the **anova**).

```
anova, regress
```

```
  Source |       SS       df       MS                  Number of obs =      20
---------+------------------------------                F(  3,    16) =    3.26
   Model |       26.15     3  8.71666667                Prob > F      =  0.0492
Residual |       42.80    16       2.675                R-squared     =  0.3793
---------+------------------------------                Adj R-squared =  0.2629
   Total |       68.95    19  3.62894737                Root MSE      =  1.6355

------------------------------------------------------------------------------
       y        Coef.   Std. Err.       t     P>|t|       [95% Conf. Interval]
------------------------------------------------------------------------------
_cons             6.4   .7314369      8.750   0.000       4.849423    7.950577
x
       1           -2   1.034408     -1.933   0.071      -4.192847    .1928471
       2           -3   1.034408     -2.900   0.010      -5.192847   -.8071529
       3         -.8    1.034408     -0.773   0.451      -2.992847    1.392847
       4    (dropped)
------------------------------------------------------------------------------
```

From the underlying regression table, notice that **anova** dropped the fourth level of **x** (the **noconstant** option of **anova** could have been used to exclude the constant if desired). In our case, the mean of the dropped level (four) is equal to the coefficient for the constant. The mean of the other three levels is equal to the coefficient for that level plus the coefficient for the constant.

After **regress** you could say something like **test cat1 = cat2** because there is a one to one correspondence between variable names and coefficients. After **anova** you must say something like **test _b[x[1]] = _b[x[2]]** (that is include terms inside of **_b[]** or the synonym **_coef[]** and indicate the level of the term). This is because with **anova** a variable corresponds to more than one coefficient. (After **regress** we could have also used the **_b[]** notation, but most people do not.)

Another helpful piece of information is that by definition the coefficient for a dropped level is zero. The following table illustrates how the coefficients from the **anova** above are related to the mean for each level of **x**.

Mean of x	anova coefficients
level 1	_b[x[1]]+_b[_cons]
level 2	_b[x[2]]+_b[_cons]
level 3	_b[x[3]]+_b[_cons]
level 4	_b[_cons]

The **test** commands for obtaining our three example tests after the latest **anova** are shown below:

1. test level one against level two

```
test _b[x[1]] = _b[x[2]]

( 1)  x[1] - x[2] = 0.0

      F(  1,    16) =    0.93
           Prob > F =    0.3481
```

2. test the average of level one and two against level three

```
test 0.5*_b[x[1]] + 0.5*_b[x[2]] = _b[x[3]]

( 1)  .5 x[1] + .5 x[2] - x[3] = 0.0

      F(  1,    16) =    3.60
           Prob > F =    0.0759
```

3. test that five times level one plus four times level two minus three times level three equals two times level four

```
test 5*(_b[x[1]]+_b[_cons]) + 4*(_b[x[2]]+_b[_cons]) - 3*(_b[x[3]]+
     _b[_cons]) = 2*(_b[_cons])

( 1)  4.0 _cons + 5.0 x[1] + 4.0 x[2] - 3.0 x[3] = 0.0

      F(  1,    16) =    1.25
           Prob > F =    0.2808
```

In the first two tests the **_b[_cons]** cancels out and does not need to be included in the test statement at all. In fact, the first two **test** commands are probably what you would have guessed them to be without ever having looked closely at the underlying regression model. And for many of the tests that people typically perform this will be true.

The third (and admittedly stranger) test is different. The constant does not cancel out. If you were to execute your first guess as to the proper **test** command you would not have obtained the test you desired. In this case you need to look at how the underlying coefficients relate to the quantities of interest to you. Notice that in this third test we allowed Stata's **test** command to do the algebra for us. We simply entered the appropriate linear combination of coefficients for each cell mean used in the linear combination being tested.

Two categorical factors

When there are two (or more) categorical factors in our model we again may want to test various single degree-of-freedom hypotheses that compare various levels of the two (or more) factors in the model. As with the one-way ANOVA model, the way in which you parameterize your two-way (or higher) model affects how you go about performing individual tests.

To demonstrate how to obtain single degree-of-freedom tests after a two-way ANOVA we will use the following 24 observation dataset where the variables **a** and **b** are categorical variables with 4 and 3 levels respectively and there is a response variable **y**.

```
list, noobs
```

```
  a   b   y
  1   1  26
  1   1  30
  1   2  54
  1   2  50
  1   3  34
  1   3  46
  2   1  16
  2   1  20
  2   2  36
  2   2  24
  2   3  50
  2   3  34
  3   1  48
  3   1  28
  3   2  28
  3   2  28
  3   3  50
  3   3  46
  4   1  50
  4   1  46
  4   2  48
  4   2  44
  4   3  48
  4   3  28
```

The following table shows the mean of **y** for each cell of **a** by **b** as well as the means for each level of **a** and **b** (the column and row titled "Total"):

```
table a b, c(mean y) row col
```

```
----------+---------------------------
          |             b
        a |     1      2      3  Total
----------+---------------------------
        1 |    28     52     40     40
        2 |    18     30     42     30
        3 |    38     28     48     38
        4 |    48     46     38     44
          |
    Total |    33     39     42     38
----------+---------------------------
```

This dataset is balanced (two observations per cell). If you are dealing with unbalanced data (including the case where you have missing cells) you will want to also read the Technical Notes in the section titled **Two-way analysis of variance** in the [R] **anova** manual entry.

The standard way of performing an ANOVA on this data is with

```
anova y a b a*b
```

```
                          Number of obs =      24     R-squared     =  0.7606
                          Root MSE      = 7.74597     Adj R-squared =  0.5412

                 Source |  Partial SS    df       MS           F     Prob > F
             -----------+----------------------------------------------------
                  Model |     2288.00    11      208.00       3.47     0.0214
                        |
                      a |      624.00     3      208.00       3.47     0.0509
```

```
           b |      336.00     2      168.00       2.80     0.1005
         a*b |     1328.00     6  221.333333       3.69     0.0259
             |
    Residual |      720.00    12       60.00
-------------+------------------------------------------------------
       Total |     3008.00    23  130.782609
```

This is the overparameterized two-way ANOVA model (which we will discuss in more detail later). When it comes to single degree-of-freedom tests, some people prefer to use a different parameterization – the cell means model.

Cell means ANOVA model

In the cell means ANOVA model, we first create one categorical variable that corresponds to the cells in the two-way table (or higher order table if more than two categorical variables are involved). The **egen group()** function is useful for creating the single categorical variable.

```
egen c = group(a b)

table a b, c(mean c)
```

```
----------+-----------------
          |        b
        a |    1     2     3
----------+-----------------
        1 |    1     2     3
        2 |    4     5     6
        3 |    7     8     9
        4 |   10    11    12
----------+-----------------
```

The table above reminds us how the **c** variable relates to the original **a** and **b** variables. For instance when **c** is 8 it means that **a** is 3 and **b** is 2. (If **a** and **b** had been reversed in the **egen group()** option, then the table above would show a different relationship.)

The cell means ANOVA model is then obtained by using the **noconstant** option of **anova** and the newly created **c** variable in place of **a** and **b**.

(It is, of course, also possible to use the **regress** command to perform the cell means ANOVA model. You create the full set of indicator variables corresponding to **c** and then use these along with the **noconstant** option of **regress**. Here we will instead concentrate on using the **anova** command.)

```
anova y c, nocons
```

```
                          Number of obs =      24     R-squared     =  0.9809
                          Root MSE      = 7.74597     Adj R-squared =  0.9618

                   Source |  Partial SS    df       MS           F     Prob > F
-----------------------+------------------------------------------------------
                    Model |    36944.00    12  3078.66667      51.31     0.0000
                          |
                        c |    36944.00    12  3078.66667      51.31     0.0000
                          |
                 Residual |      720.00    12       60.00
-----------------------+------------------------------------------------------
                    Total |    37664.00    24  1569.33333
```

```
anova, regress
```

```
  Source |       SS       df       MS                 Number of obs =      24
---------+------------------------------              F( 12,    12) =   51.31
   Model |    36944.00    12  3078.66667              Prob > F      =  0.0000
Residual |      720.00    12       60.00              R-squared     =  0.9809
---------+------------------------------              Adj R-squared =  0.9618
   Total |    37664.00    24  1569.33333              Root MSE      =   7.746

------------------------------------------------------------------------------
       y        Coef.   Std. Err.       t     P>|t|       [95% Conf. Interval]
------------------------------------------------------------------------------
c
       1           28   5.477226      5.112   0.000       16.06615    39.93385
       2           52   5.477226      9.494   0.000       40.06615    63.93385
       3           40   5.477226      7.303   0.000       28.06615    51.93385
       4           18   5.477226      3.286   0.007       6.066151    29.93385
       5           30   5.477226      5.477   0.000       18.06615    41.93385
       6           42   5.477226      7.668   0.000       30.06615    53.93385
       7           38   5.477226      6.938   0.000       26.06615    49.93385
       8           28   5.477226      5.112   0.000       16.06615    39.93385
```

```
          9           48   5.477226      8.764   0.000       36.06615    59.93385
         10           48   5.477226      8.764   0.000       36.06615    59.93385
         11           46   5.477226      8.398   0.000       34.06615    57.93385
         12           38   5.477226      6.938   0.000       26.06615    49.93385
------------------------------------------------------------------------------
```

Compare the 12 coefficients for **c** in the table above to the table of means presented earlier. The coefficients from the cell means ANOVA model are the cell means from the two-way table. This correspondence makes creating meaningful **test** statements easy.

You can recreate an F-test from the overparameterized ANOVA model using appropriate combinations of single degree-of-freedom tests after the cell means ANOVA model. For example, the test for the term **a** with 3 degrees-of-freedom can be obtained by accumulating 3 single degree-of-freedom tests. Below I combine the tests of level 1 versus 2, level 1 versus 3, and level 1 versus 4 of the **a** variable. (Remember that **c** 1, 2, and 3 correspond to level 1 of **a**, **c** 4, 5, and 6 correspond to level 2 of **a**, and so on.)

```
test _b[c[1]] + _b[c[2]] + _b[c[3]] = _b[c[4]] + _b[c[5]] + _b[c[6]]

 ( 1)  c[1] + c[2] + c[3] - c[4] - c[5] - c[6] = 0.0

       F(  1,    12) =    5.00
            Prob > F =    0.0451

test _b[c[1]] + _b[c[2]] + _b[c[3]] = _b[c[7]] + _b[c[8]] + _b[c[9]], accum

 ( 1)  c[1] + c[2] + c[3] - c[4] - c[5] - c[6] = 0.0
 ( 2)  c[1] + c[2] + c[3] - c[7] - c[8] - c[9] = 0.0

       F(  2,    12) =    2.80
            Prob > F =    0.1005

test _b[c[1]] + _b[c[2]] + _b[c[3]] = _b[c[10]] + _b[c[11]] + _b[c[12]], accum

 ( 1)  c[1] + c[2] + c[3] - c[4] - c[5] - c[6] = 0.0
 ( 2)  c[1] + c[2] + c[3] - c[7] - c[8] - c[9] = 0.0
 ( 3)  c[1] + c[2] + c[3] - c[10] - c[11] - c[12] = 0.0

       F(  3,    12) =    3.47
            Prob > F =    0.0509
```

This F of 3.47 agrees with the F-test for the term **a** in the over-parameterized ANOVA presented earlier. Of course, it is easier to obtain the test of a term like **a** by running the over-parameterized model. I just wanted to show that you could also obtain the result with a little work starting from the cell means model.

For completeness sake, here is a set of **test** commands to produce the F-test for the **b** term.

```
test _b[c[1]]+_b[c[4]]+_b[c[7]]+_b[c[10]]=_b[c[2]]+_b[c[5]]+_b[c[8]]+_b[c[11]]
test _b[c[1]]+_b[c[4]]+_b[c[7]]+_b[c[10]]=_b[c[3]]+_b[c[6]]+_b[c[9]]+_b[c[12]], accum
```

Here is a set of **test** commands to produce the F-test for the **a** by **b** interaction term.

```
test _b[c[1]] + _b[c[5]] = _b[c[2]] + _b[c[4]]
test _b[c[1]] + _b[c[6]] = _b[c[3]] + _b[c[4]], accum
test _b[c[1]] + _b[c[8]] = _b[c[2]] + _b[c[7]], accum
test _b[c[1]] + _b[c[9]] = _b[c[3]] + _b[c[7]], accum
test _b[c[1]] + _b[c[11]] = _b[c[2]] + _b[c[10]], accum
test _b[c[1]] + _b[c[12]] = _b[c[3]] + _b[c[10]], accum
```

There are actually many different ways I could have combined various single degree-of-freedom tests to obtain the overall F-tests for the **a**, **b**, and **a** by **b** terms.

Now that we have demonstrated that you can reproduce the results from the over-parameterized model with an appropriate series of **test** statements after a cell means model, let us now look at a few different single degree-of-freedom tests. (We will later see how to obtain these same single degree-of-freedom tests after the over-parameterized ANOVA.) You will want to look back at the table showing how **c** relates to **a** and **b** to see how these tests were constructed.

1. test level two against level four of factor **a**

```
test _b[c[4]] + _b[c[5]] + _b[c[6]] = _b[c[10]] + _b[c[11]] + _b[c[12]]

( 1)  c[4] + c[5] + c[6] - c[10] - c[11] - c[12] = 0.0

      F(  1,    12) =    9.80
           Prob > F =    0.0087
```

2. test the average of level one and two against level three of factor **b**

```
test (_b[c[1]] + _b[c[4]] + _b[c[7]] + _b[c[10]] + _b[c[2]] + _b[c[5]] +
      _b[c[8]] + _b[c[11]])/2 = _b[c[3]] + _b[c[6]] + _b[c[9]] + _b[c[12]]

( 1)  .5 c[1] + .5 c[2] - c[3] + .5 c[4] + .5 c[5] - c[6] + .5 c[7] + .5 c[8]
      - c[9] + .5 c[10] + .5 c[11] - c[12] = 0.0

      F(  1,    12) =    3.20
           Prob > F =    0.0989
```

3. test the average of level one and two of **a**, when **b** is also at level one or two, against the average of level three and four of **a**, when **b** is at level three

```
test (_b[c[1]] + _b[c[4]] + _b[c[2]] + _b[c[5]])/2 = _b[c[9]] + _b[c[12]]

( 1)  .5 c[1] + .5 c[2] + .5 c[4] + .5 c[5] - c[9] - c[12] = 0.0

      F(  1,    12) =    5.38
           Prob > F =    0.0388
```

4. test that three times **a** at one and **b** at one, minus four times **a** at three and **b** at two, plus six times **a** at four and **b** at three, equals **a** at two and **b** at two, minus two times **a** at two and **b** at three

```
test 3*_b[c[1]] - 4*_b[c[8]] + 6*_b[c[12]] = _b[c[5]] - 2*_b[c[6]]

( 1)  3.0 c[1] - c[5] + 2.0 c[6] - 4.0 c[8] + 6.0 c[12] = 0.0

      F(  1,    12) =   32.58
           Prob > F =    0.0001
```

Constructing various single degree-of-freedom tests after a cell means ANOVA model is relatively easy. You pick the appropriate linear combination of the coefficients based on how the single categorical variable (**c** in our example) relates to the original categorical variables (**a** and **b** in our example) and based on the hypothesis of interest.

Overparameterized ANOVA model

Most people are used to the results presented by the overparameterized ANOVA model. As we saw when we discussed the cell means ANOVA model, the F-tests for terms in the ANOVA model are obtained directly from the overparameterized model ANOVA table. Compare this to computing an F-test for a term by accumulating the results of several individual degree-of-freedom tests after the cell means ANOVA model. However, when it comes to obtaining single degree-of-freedom tests, most people find the cell means model approach to be the easiest.

Here, again, is the overparameterized ANOVA model for our example data. In addition, I use the **regress** command to replay the ANOVA as a regression table (I could have also said **anova, regress** to see this display). When you see the various levels reported as "dropped", you begin to understand why it is called the overparameterized ANOVA model.

```
anova y a b a*b

                          Number of obs =      24     R-squared     =  0.7606
                          Root MSE      = 7.74597     Adj R-squared =  0.5412

                   Source |  Partial SS    df       MS           F     Prob > F
               -----------+----------------------------------------------------
                    Model |     2288.00    11      208.00       3.47     0.0214
                          |
                        a |      624.00     3      208.00       3.47     0.0509
                        b |      336.00     2      168.00       2.80     0.1005
                      a*b |     1328.00     6  221.333333       3.69     0.0259
                          |
                 Residual |      720.00    12       60.00
               -----------+----------------------------------------------------
                    Total |     3008.00    23  130.782609

regress, noheader

------------------------------------------------------------------------------
           y        Coef.   Std. Err.       t     P>|t|       [95% Conf. Interval]
------------------------------------------------------------------------------
       _cons           38   5.477226      6.938   0.000       26.06615    49.93385
       a
           1            2   7.745967      0.258   0.801      -14.87701    18.87701
           2            4   7.745967      0.516   0.615      -12.87701    20.87701
           3           10   7.745967      1.291   0.221      -6.877011    26.87701
           4    (dropped)
       b
```

```
          1           10   7.745967      1.291   0.221      -6.877011    26.87701
          2            8   7.745967      1.033   0.322      -8.877011    24.87701
          3    (dropped)
a*b
    1  1          -22   10.95445     -2.008   0.068       -45.8677    1.867698
    1  2            4   10.95445      0.365   0.721       -19.8677     27.8677
    1  3    (dropped)
    2  1          -34   10.95445     -3.104   0.009       -57.8677    -10.1323
    2  2          -20   10.95445     -1.826   0.093       -43.8677    3.867698
    2  3    (dropped)
    3  1          -20   10.95445     -1.826   0.093       -43.8677    3.867698
    3  2          -28   10.95445     -2.556   0.025       -51.8677   -4.132302
    3  3    (dropped)
    4  1    (dropped)
    4  2    (dropped)
    4  3    (dropped)
------------------------------------------------------------------------------
```

Now the important question is how the coefficients in this model relate to the cell means. To refresh your memory, here is the table of cell means (and marginal means).

```
table a b, c(mean y) row col
```

```
          ----------+---------------------------
                    |             b
                  a |     1      2      3  Total
          ----------+---------------------------
                  1 |    28     52     40     40
                  2 |    18     30     42     30
                  3 |    38     28     48     38
                  4 |    48     46     38     44
                    |
              Total |    33     39     42     38
          ----------+---------------------------
```

The cell mean for level i of **a** and level j of **b** is equal to the coefficient for the constant plus the coefficient for **a** at level i plus the coefficient for **b** at level j plus the coefficient for **a** and **b** at i and j. When a coefficient is dropped in the regression table, the corresponding coefficient is zero. The table below shows the relationship.

a	b	cell mean	cell mean (simplified)
a = 1	b = 1	_b[_cons]+_b[a[1]]+_b[b[1]]+_b[a[1]*b[1]]	_b[_cons]+_b[a[1]]+_b[b[1]]+_b[a[1]*b[1]]
a = 1	b = 2	_b[_cons]+_b[a[1]]+_b[b[2]]+_b[a[1]*b[2]]	_b[_cons]+_b[a[1]]+_b[b[2]]+_b[a[1]*b[2]]
a = 1	b = 3	_b[_cons]+_b[a[1]]+_b[b[3]]+_b[a[1]*b[3]]	_b[_cons]+_b[a[1]]
a = 2	b = 1	_b[_cons]+_b[a[2]]+_b[b[1]]+_b[a[2]*b[1]]	_b[_cons]+_b[a[2]]+_b[b[1]]+_b[a[2]*b[1]]
a = 2	b = 2	_b[_cons]+_b[a[2]]+_b[b[2]]+_b[a[2]*b[2]]	_b[_cons]+_b[a[2]]+_b[b[2]]+_b[a[2]*b[2]]
a = 2	b = 3	_b[_cons]+_b[a[2]]+_b[b[3]]+_b[a[2]*b[3]]	_b[_cons]+_b[a[2]]
a = 3	b = 1	_b[_cons]+_b[a[3]]+_b[b[1]]+_b[a[3]*b[1]]	_b[_cons]+_b[a[3]]+_b[b[1]]+_b[a[3]*b[1]]
a = 3	b = 2	_b[_cons]+_b[a[3]]+_b[b[2]]+_b[a[3]*b[2]]	_b[_cons]+_b[a[3]]+_b[b[2]]+_b[a[3]*b[2]]
a = 3	b = 3	_b[_cons]+_b[a[3]]+_b[b[3]]+_b[a[3]*b[3]]	_b[_cons]+_b[a[3]]
a = 4	b = 1	_b[_cons]+_b[a[4]]+_b[b[1]]+_b[a[4]*b[1]]	_b[_cons]+_b[b[1]]
a = 4	b = 2	_b[_cons]+_b[a[4]]+_b[b[2]]+_b[a[4]*b[2]]	_b[_cons]+_b[b[2]]
a = 4	b = 3	_b[_cons]+_b[a[4]]+_b[b[3]]+_b[a[4]*b[3]]	_b[_cons]

The simplifications shown at the far right of the table are due to the coefficients that are dropped from the overparameterized model being zero. The marginal means can easily be built up by averaging appropriate cell means together.

We can obtain the same four, single degree-of-freedom, tests as were obtained with the cell means ANOVA model by examining the relationship (shown in the table above) between the coefficients of the overparameterized model and the quantities of real interest – the cell means. We could simply plug in all the coefficients for each cell involved in the test and let Stata's **test** command do the algebra, or we can do the simplifying ourselves. For the same four tests that were performed for the cell means model I will show the results when you plug everything into **test** (based on the simplification in the far right column of the table above) and let Stata's **test** command do the algebra. (It also would work if I plugged the unsimplified cell mean expressions into **test**.)

1. test level two against level four of factor **a**

```
test (_b[_cons]+_b[a[2]]+_b[b[1]]+_b[a[2]*b[1]]) + (_b[_cons]+_b[a[2]]+
     _b[b[2]]+_b[a[2]*b[2]]) + (_b[_cons]+_b[a[2]]) = (_b[_cons]+_b[b[1]]) +
     (_b[_cons]+_b[b[2]]) + (_b[_cons])
```

```
( 1)  3.0 a[2] + a[2]*b[1] + a[2]*b[2] = 0.0
```

```
       F(  1,    12) =    9.80
            Prob > F =    0.0087
```

2. test the average of level one and two against level three of factor **b**

```
test ((_b[_cons]+_b[a[1]]+_b[b[1]]+_b[a[1]*b[1]]) + (_b[_cons]+_b[a[2]]+
      _b[b[1]]+_b[a[2]*b[1]]) + (_b[_cons]+_b[a[3]]+_b[b[1]]+_b[a[3]*b[1]]) +
      (_b[_cons]+_b[b[1]]) + (_b[_cons]+_b[a[1]]+_b[b[2]]+_b[a[1]*b[2]]) +
      (_b[_cons]+_b[a[2]]+_b[b[2]]+_b[a[2]*b[2]]) + (_b[_cons]+_b[a[3]]+
      _b[b[2]]+_b[a[3]*b[2]]) + (_b[_cons]+_b[b[2]]))/2 = (_b[_cons]+_b[a[1]])
      + (_b[_cons]+_b[a[2]]) + (_b[_cons]+_b[a[3]]) + (_b[_cons])

 ( 1)  2.0 b[1] + 2.0 b[2] + .5 a[1]*b[1] + .5 a[1]*b[2] + .5 a[2]*b[1] +
5 a[2]*b[2] + .5 a[3]*b[1] + .5 a[3]*b[2] = 0.0

       F(  1,    12) =    3.20
            Prob > F =    0.0989
```

3. test the average of level one and two of **a**, when **b** is also at level one or two, against the average of level three and four of **a**, when **b** is at level three

```
test ((_b[_cons]+_b[a[1]]+_b[b[1]]+_b[a[1]*b[1]]) + (_b[_cons]+_b[a[2]]+
      _b[b[1]]+_b[a[2]*b[1]]) + (_b[_cons]+_b[a[1]]+_b[b[2]]+_b[a[1]*b[2]]) +
      (_b[_cons]+_b[a[2]]+_b[b[2]]+_b[a[2]*b[2]]))/2 = (_b[_cons]+_b[a[3]]) +
      (_b[_cons])

 ( 1)  a[1] + a[2] - a[3] + b[1] + b[2] + .5 a[1]*b[1] + .5 a[1]*b[2] +
5 a[2]*b[1] + .5 a[2]*b[2] = 0.0

       F(  1,    12) =    5.38
            Prob > F =    0.0388
```

4. test that three times **a** at one and **b** at one, minus four times **a** at three and **b** at two, plus six times **a** at four and **b** at three, equals **a** at two and **b** at two, minus two times **a** at two and **b** at three

```
test 3*(_b[_cons]+_b[a[1]]+_b[b[1]]+_b[a[1]*b[1]]) - 4*(_b[_cons]+_b[a[3]]+
      _b[b[2]]+_b[a[3]*b[2]]) + 6*(_b[_cons]) = (_b[_cons]+_b[a[2]]+_b[b[2]]+
      _b[a[2]*b[2]]) - 2*(_b[_cons]+_b[a[2]])

 ( 1)  6.0 _cons + 3.0 a[1] + a[2] - 4.0 a[3] + 3.0 b[1] - 5.0 b[2] +
       3.0 a[1]*b[1] - a[2]*b[2] - 4.0 a[3]*b[2] = 0.0

       F(  1,    12) =   32.58
            Prob > F =    0.0001
```

You can compare these results to those obtained after the cell means ANOVA model to see that they are the same.

For More Information

- Stata FAQ- How can I do ANOVA contrasts in Stata?
- Stata FAQ- How can I create dummy variables in Stata?
