The Stata Journal

The *Stata Journal* publishes reviewed papers together with shorter notes or comments, regular columns, book reviews, and other material of interest to Stata users. Examples of the types of papers include 1) expository papers that link the use of Stata commands or programs to associated principles, such as those that will serve as tutorials for users first encountering a new field of statistics or a major new technique; 2) papers that go "beyond the Stata manual" in explaining key features or uses of Stata that are of interest to intermediate or advanced users of Stata; 3) papers that discuss new commands or Stata programs of interest either to a wide spectrum of users (e.g., in data management or graphics) or to some large segment of Stata users (e.g., in survey statistics, survival analysis, panel analysis, or limited dependent variable modeling); 4) papers analyzing the statistical properties of new or existing estimators and tests in Stata; 5) papers that could be of interest or usefulness to researchers, especially in fields that are of practical importance but are not often included in texts or other journals, such as the use of Stata in managing datasets, especially large datasets, with advice from hard-won experience; and 6) papers of interest to those who teach, including Stata with topics such as extended examples of techniques and interpretation of results, simulations of statistical concepts, and overviews of subject areas.

For more information on the *Stata Journal*, including information for authors, see the webpage

http://www.stata-journal.com

The *Stata Journal* is indexed and abstracted in the following:

- CompuMath Citation Index®
- Current Contents/Social and Behavioral Sciences®
- RePEc: Research Papers in Economics
- Science Citation Index Expanded (also known as SciSearch®)
- Scopus™
- Social Sciences Citation Index®

The *Stata Journal*, electronic version (ISSN 1536-8734) is a publication of Stata Press. Stata, Stata, Stata Press, Mata, Mata, and NetCourse are registered trademarks of StataCorp LP.

The Stata Journal (2012)
12, Number 2, pp. 342–344

Stata tip 108: On adding and constraining

Maarten L. Buis
Department of Sociology
Tübingen University
Tübingen, Germany
maarten.buis@uni-tuebingen.de

In many estimation commands, the `constraint` command (see [R] **constraint**) can impose linear constraints. The most common of these is the constraint that two or more regression coefficients are equal. A sometimes useful characteristic of models with that constraint is that they are equivalent to a model that includes the sum of the variables that are constrained. Consider the relevant part of a regression equation:

$$\beta_1 x_1 + \beta_2 x_2$$

If we constrain the effects of x_1 and x_2 to be equal, then we can replace β_1 and β_2 with β:

$$\beta x_1 + \beta x_2 = \beta(x_1 + x_2)$$

One situation where this characteristic can be useful occurs when you have created a variable by adding several variables and you wonder whether that was a good idea. In the example below, there are three variables on the degree of trust a respondent has in the executive, legislative, and judicial branches of the U.S. federal government: `confed`, `conlegis`, and `conjudge`, respectively. These can take the values 0 (hardly any trust), 1 (only some trust), or 2 (a great deal of trust). I think that these three variables say something about the trust in the federal government, and I created a single variable that captures that, `congov`, which I use to predict whether a respondent voted for Barack Obama in the 2008 U.S. presidential election. This results in model `sum1`.

If I want to check whether adding these three confidence measures was a good idea, I can use the fact that adding variables is equivalent to constraining their effects to be equal. So you can operationalize the rather vague idea "adding these variables is a good idea" to the testable statement "the effects of these three variables are the same". As a check, I first fit a model that constrains the effects to be equal. This is model `constr1`, and as expected, the resulting coefficients, standard errors, and log likelihood are exactly the same. I then fit a model with the three confidence variables without constraint, `unconstr1`. The resulting coefficients are very different from one another: the effects do not even have the same sign.[1] A likelihood-ratio test also rejects the hypothesis that these variables have the same effect on voting for Obama. So adding the sum of the three confidence measures was not a good idea in this case.

1. These are odds ratios, so the sign is determined by whether the ratio is larger or smaller than 1.

 st0261

```
. use gss10
(extract from the 2010 General Social Survey)
. generate byte congov = confed + conlegis + conjudge
(463 missing values generated)
. quietly logit obama congov, or nolog
. estimates store sum1
. constraint 1 confed = conlegis
. constraint 2 confed = conjudge
. quietly logit obama confed conlegis conjudge, or constraint(1 2) nolog
. estimates store constr1
. quietly logit obama confed conlegis conjudge, or nolog
. estimates store unconstr1
. estimates table sum1 constr1 unconstr1, stats(ll N) eform b(%9.3g) se(%9.3g)
> stfmt(%9.4g)
```

Variable	sum1	constr1	unconstr1
congov	1.62		
	.11		
confed		1.62	3.47
		.11	.576
conlegis		1.62	1.69
		.11	.305
conjudge		1.62	.674
		.11	.107
_cons	.461	.461	.689
	.0833	.0833	.134
ll	-347.8	-347.8	-324.9
N	557	557	557

```
                                                  legend: b/se
. lrtest constr1 unconstr1
Likelihood-ratio test                                 LR chi2(2)  =     45.77
(Assumption: constr1 nested in unconstr1)             Prob > chi2 =    0.0000
```

Another situation where this characteristic can be useful occurs when you have two or more ordinal or categorical variables that you want to combine. Consider the example below. In that example, I want to treat education as an ordinal variable, and I want to see the effect of "family educational background" on the educational attainment of the children.

I think of family educational background as some sort of sum of the father's and mother's educations, but how do I create a sum of two ordinal variables? That is hard, but it is easy to consider the equivalent model that constrains the effects of father's education to be equal to the effects of mother's education. In this example, the effects of mother's and father's educations are fairly similar, and the test of the hypothesis that they are equal cannot be rejected (compare `unconstr2` with `constr2`). It also shows that constraining effects to be the same is equivalent to adding the sums of the indicator variables (compare `constr2` with `sum2`).

```
. quietly ologit degree i.madeg i.padeg, or nolog
. estimates store unconstr2
. constraint 1 1.madeg = 1.padeg
. constraint 2 2.madeg = 2.padeg
. quietly ologit degree i.madeg i.padeg, or constraint(1 2) nolog
. estimates store constr2
. generate byte p_hs = 1.madeg + 1.padeg
(329 missing values generated)
. generate byte p_mths = 2.madeg + 2.padeg
(329 missing values generated)
. quietly ologit degree p_hs p_mths, or nolog
. estimates store sum2
. estimates table unconstr2 constr2 sum2, stats(ll N) eform b(%9.3g) se(%9.3g)
> stfmt(%9.4g) keep(degree:)

--------------------------------------------------
    Variable | unconstr2    constr2       sum2
-------------+------------------------------------
       madeg |
           1 |       2.5        2.17
             |      .449        .199
           2 |      4.89        5.51
             |      1.26        .691
             |
       padeg |
           1 |      1.88        2.17
             |      .322        .199
           2 |      6.08        5.51
             |      1.46        .691
             |
        p_hs |                              2.17
             |                              .199
      p_mths |                              5.51
             |                              .691
-------------+------------------------------------
          ll |    -799.1      -800.5      -800.5
           N |       972         972         972
--------------------------------------------------
                                      legend: b/se

. lrtest unconstr2 constr2
Likelihood-ratio test                                 LR chi2(2)  =      2.79
(Assumption: constr2 nested in unconstr2)             Prob > chi2 =    0.2484
```