Econ 452 Section 4 - STATA

Connor Cole

October 5, 2015

Basic Statistical Tools, Continued

Chi-Squared Tests and Frequency Tables

Again, the document available on Canvas should describe how chi-squared tests work. Chi-squared tests with tables assess whether or not status in a categorical variable seems to systematically differ with characteristics in another categorical variable. For example, with our data, we could use this fact to assess whether delinquency ratings seem to systematically differ by sex. To accomplish this test, we would type:

```
tabulate variablename1 variablename2, chi2
```

To get a sense of how rates differ by variable category, there are other options we can add to the `tabulate` command that will describe the fraction of observations in the column. To see the fraction of observations in each column by row where percentages across rows add to 100%, we type:

```
tabulate variablename1 variablename2, row
```

To see the fraction of observations in each column by row where percentages across columns add to 100%, we type:

```
tabulate variablename1 variablename2, col
```

To see the fraction of observations in each cell where percentages across all cells add to 100%, we type:

```
tabulate variablename1 variablename2, cell
```

Note that we can include all commands together for one grand table of percentages and a Chi-squared test:

```
tabulate variablename1 variablename2, chi2 row col cell
```

More Advanced Data Management

STATA has many command shortcuts that make some of the conditioning arguments we have considered so far much easier. Here are some of the most important ones you might use:

The 'Sort' Command

The orders of rows in STATA can be reorganized easily. If you're wanting to browse specific values of certain subgroups, this command will be helpful, but in general as you become more proficient at using STATA without having to look at the data you will find yourself using it less. If we wanted to organize the data by some variable, we would type:

```
sort variablename1
```

Note that sorting on a variable just changes around the order of rows and does not affect the composition of individual observations. STATA will sort by default using 'ascending' values. Every variable in the dataset we use is numeric, so STATA will sort by ascending values in the category of choice. If we had string data, or data where the observations are saved as text without an underlying number beneath them in the dataset, STATA would organize observations alphabetically.

If we wanted to sort by one variable, and then sort the observations in each category by another variable, we would type:

```
sort variablename1 variablename2
```

The 'Bysort' Command

Suppose you wanted to calculate the mean of some variable, say income, by different values of a categorical variable, and let's assume that the categorical variable takes on only two values, 0 and 1. We already know that we can see the mean of this variable for these different categories by using a conditioning argument for both conditions: `summary variablename1 if variablename2==0` `summary variablename1 if variablename2==1`

However, this process may become unwieldy quickly if there are more categories present. In general, if we want to either compute statistics or create new variables using the This command organizes the data using `sort` and then does whatever operation we have told STATA to do by the different categories created by whatever variable(s) by which we have organized the data. So, returning to the previous example, we could run the same calculations with:

```
bysort variablename2:  summary variablename1
```

In addition to computing statistics, the `bysort` command is also useful when used with the `egen` to calculate functions of variables in a column. For example, if we wanted to create a new

variable recording the relevant mean of the first variable for the group that the observations belongs to in the second variable, we would type:

```
bysort:  variablename2:  egen groupmeans=mean(variablename1)
```

Note that like we did using the `sort` command, we can easily add more categorical variables to create smaller subgroups:

```
bysort variablename2 variablename3:  summary variablename1
```

The 'Inrange' Command

Suppose we wanted to compute statistics for some subgroup defined by a range of values of some other variable. Again, we could do this long form by putting all the conditioning arguments together:

```
summarize variablename1 if variablename==2 | variablename2==3 | variablename2==4
```

However, a much more efficient way of coding this command would be to write:

```
summarize variablename1 if inrange(variablename2,2,4)
```

Note that the `inrange` command will include all observations that have variable values that fall in the range we've established unless we purposely exclude them.

The 'Inlist' Command

Suppose we wanted to compute statistics for some subgroup defined by a non-adjacent range of values of some other variable. Again, we could do this long form by putting all the conditioning arguments together:

```
summarize variablename1 if variablename==2 | variablename2==4 | variablename2==9
```

Again, a much more efficient way would be to write:

```
summarize variablename1 if inlist(variablename2,2,4,9)
```

Regression

Estimating a Regression

Running regressions in STATA is very straight-forward. We simply type in `regression`, then type the variable we want as a dependent variable, or the variable we are predicting, and then type after the dependent variable the variables that we want to include as covariates to predict the dependent variable. Therefore, if we wanted to estimate the following equation:

$$VariableName1_i = \beta_0 + \beta_1 \cdot VariableName2_i + u_i$$

We would type:

```
regress VariableName1 VariableName2
```

If we wanted to include additional covariates and estimate a multi-variate regression, say of the form:

$$VariableName1_i = \beta_0 + \beta_1 \cdot VariableName2_i + \beta_2 \cdot VariableName3_i + u_i$$

We would type:

```
regress VariableName1 VariableName2 VariableName3
```

While telling STATA to estimate parameters from a regression is straight-forward, as with the t-tests before, interpreting the different parts of the output from a regression can be tricky. Let's say we wanted to regress information about pre-tax earnings on scores on a standardized test, the ASVAB.

Consider the output from STATA below:

Here are the components of STATA's output:

1. Sum of squares calculations. Here STATA decomposes the sum of squared deviations of the dependent variable, `pretaxlaborincome` in this case, from its average into the both explained sum of squares (`Model`) and the residual sum of squares (`Residual`). Remember:

$$\text{Total Sum of Squares} = \text{Model Sum of Squares} + \text{Residual Sum of Squares}$$
$$\sum_{i=1}^{n}(y_i - \bar{y})^2 = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2$$

Figure 1: Regression Output

```
. reg pretaxlaborincome asvabmathverbal_1999

      Source |       SS           df       MS      Number of obs   =     5,516
-------------+----------------------------------   F(1, 5514)      =    755.27
       Model |  4.9528e+11         1  4.9528e+11   Prob > F        =    0.0000
    Residual |  3.6159e+12     5,514   655770269   R-squared       =    0.1205
-------------+----------------------------------   Adj R-squared   =    0.1203
       Total |  4.1112e+12     5,515   745457975   Root MSE        =     25608

------------------------------------------------------------------------------
pretaxlabo~e |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
asvabmathv~9 |   .3223284   .0117286    27.48   0.000     .2993356    .3453211
       _cons |   12779.37   638.8523    20.00   0.000     11526.97    14031.78
------------------------------------------------------------------------------
```

1

2

3

Note that STATA also reports the degrees of freedom, which will be the number of regression coefficients estimated for the model, and the sample size minus the number of coefficients estimated for the residual. Lastly, STATA reports the mean sum of squares, or the sum of squares divided by the relevant degrees of freedom.

2. Information about the fit of the whole regression. Here STATA reports the sample size, an F-test for significance of the whole regression by testing the null hypothesis that all parameter values are 0, the related p-value for the F-test, the R-squared value (which we could calculate by dividing the model sum of squares by the total sum of squares), the 'adjusted' R-squared value, or a value of the R-squared where an additional term reduces R-squared for an increased number of a variables included in the regression, and lastly the root MSE, or the sample standard deviation of the error term.

3. Parameters from the regression. STATA describes coefficients, standard errors, t-statistics, p-values and 95% confidence intervals for all covariates in the regression, including the constant term.

Note that the regressions we are running here report coefficients using the estimation procedure of least squares. STATA cannot tell us whether or not the model we are running is 'true' in some sense, and cannot tell us whether or not the conditions necessary for these estimated coefficients to be unbiased and consistent hold. That is, STATA cannot tell us that the 'true' model is linear in parameters (SLR1, MLR1 in the Wooldridge textbook), whether or not the data are randomly sampled (SLR2, MLR2), and, most importantly, whether or not $\mathbb{E}[u|X] = 0$, or whether the error term u has an expectation of 0 conditional on the covariates X (SLR 4, MLR4). These are assumptions that we need to defend independently of estimating parameters when presenting results.

However, there is one set of assumptions that we make in estimating parameters for a regression model that STATA will tell us may or may not be violated. If two variables take on the exact same

values or the same values scaled by a constant, STATA will tell us that the variables are collinear and will drop one of the variables in reporting estimated coefficients for the model. If there is no variation in a variable at all in the covariates, that is if it is collinear with the constant, STATA will drop the variable we have included. Lastly, if there is no variation in the dependent variable, then STATA will not estimate the model at all. That, is, STATA will tell us if SLR 3 and MLR 3 are met, and will either change the model to make that assumptions fit, or report if it is impossible to estimate.

Lastly, note that STATA assumes, unless told otherwise, that the error terms are homoskedastic, or that the variance of the error term is constant for all observations. In our class, we have called this assumption SLR 5 and MLR 5. Again, as previously, STATA cannot tell us whether or not this assumption is valid, and it would be up to us to defend the validity of this claim when we present our estimates of the model. Remember that this assumption is only relevant for calculating standard errors, and would not affect estimation of parameters.

As a final point, regression, like every command in STATA, can be restricted to a subgroup of individuals if we add a conditioning statement after the relevant regression command. This conditioning statement will limit the data used for estimation of the regression to observations for which the conditioning statement holds. For example, if we wanted to estimate the previous multivariate regression for observations for which a fourth variable takes on value 1, we would type:

```
regress VariableName1 VariableName2 VariableName3 if VariableName4==1
```