# Econ 452 Section 11 - STATA

Connor Cole

November 20, 2015

## Adjusted $R^2$ (Or $\overline{R}^2$)

As we discussed in class, $R^2$ is:

$$\begin{aligned} R^2 &= \frac{\sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \\ &= 1 - \frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \end{aligned}$$

Or:

$$\begin{aligned} R^2 &= \frac{SSM}{SST} \\ &= 1 - \frac{SSE}{SST} \end{aligned}$$

Adjusted $R^2$, or $\overline{R}^2$ can be computed as:

$$\begin{aligned} \overline{R}^2 &= \frac{\frac{\sum_{i=1}^{n}(y_i-\bar{y})^2}{n-1} - \frac{\sum_{i=1}^{n}(\hat{y}_i-\bar{y})^2}{n-k}}{\frac{\sum_{i=1}^{n}(y_i-\bar{y})^2}{n-1}} \\ &= 1 - \frac{\frac{\sum_{i=1}^{n}(\hat{y}_i-y_i)^2}{n-k}}{\frac{\sum_{i=1}^{n}(y_i-\bar{y})^2}{n-1}} \end{aligned}$$

Or:

$$\begin{aligned}\overline{R}^2 &= \frac{\frac{SST}{n-1} - \frac{SSM}{n-k}}{\frac{SST}{n-1}} \\ &= 1 - \frac{\frac{SSM}{n-k}}{\frac{SST}{n-1}}\end{aligned}$$

As you know, $R^2$ always increases with additional covariates, which makes it a bad tool for evaluating the descriptiveness of a particular regression model compared to another, especially when higher order terms are included. This fact motivates $\overline{R}^2$, a measure of descriptiveness that actually decreases if additional covariates don't contribute 'enough' descriptive power to the model (technically, an additional covariate increases $\overline{R}^2$ if and only if the $t$ statistic under the null hypothesis that the 'true' value of $\beta_k$ is 0, or $\frac{\hat{\beta}_k - 0}{se(\hat{\beta}_k)}$ is greater than or equal to 1.

Consider the output from a regression of $VariableName1$ on $VariableName2$ and $VariableName3$:

Figure 1: Adjusted $R^2$ in a Regression

```
. regress VariableName1 VariableName2 VariableName3
```

| Source | SS | df | MS | Number of obs | = | 1,085 |
|---|---|---|---|---|---|---|
| | | | | F(2, 1082) | = | 94.21 |
| Model | 1.1612e+11 | 2 | 5.8061e+10 | Prob > F | = | 0.0000 |
| Residual | 6.6680e+11 | 1,082 | 616265056 | R-squared | = | 0.1483 |
| | | | | Adj R-squared | = | 0.1467 |
| Total | 7.8292e+11 | 1,084 | 722252292 | Root MSE | = | 24825 |

| VariableName1 | Coef. | Std. Err. | t | P>\|t\| | [95% Conf. | Interval] |
|---|---|---|---|---|---|---|
| VariableName2 | 2151.093 | 323.8249 | 6.64 | 0.000 | 1515.697 | 2786.489 |
| VariableName3 | .180467 | .0316651 | 5.70 | 0.000 | .118335 | .242599 |
| _cons | -10696.45 | 3846.24 | -2.78 | 0.006 | -18243.39 | -3149.516 |

In the far right table, we see both $R^2$ and Adjusted R squared, which is the same as $\overline{R}^2$. We can rederive these results using our formulas:

$$
\begin{aligned}
R^2 &= 1 - \frac{SSE}{SST} \\
&= 1 - \frac{6.6680e+11}{7.8292e+11} = 0.1483 \\
\overline{R}^2 &= 1 - \frac{\frac{SSE}{n-k-1}}{\frac{SST}{n-1}} \\
&= 1 - \frac{\frac{6.6680e+11}{1085-2-1}}{\frac{7.8292e+11}{1085-1}} = 0.1467
\end{aligned}
$$

## Lagrange Multiplier Test in STATA

One last hypothesis test method we covered before is the Lagrange Multiplier Test. Many facets of the Lagrange Multiplier Test are similar to the F-test that we considered before. First, we have an 'unrestricted' model where we haven't imposed our null hypotheses, and a 'restricted' model where we directly impose our null hypothesis. However, the set-up for the Lagrange Multiplier Test is a little different. To run a Lagrange Multiplier Test, we first estimate our restricted model where we directly impose our null hypothesis. We then pull residuals from this regression and then regress the residuals on the full set of covariates from the unrestricted model. Then, under the null hypothesis that the restrictions in the restricted model are true, then $n \cdot R^2$ from this second regression should be $\chi^2_q$, or have a Chi-squared distribution with $q$ degrees of freedom.

Let's consider the model below:

$$VariableName1 = \beta_0 + \beta_1 VariableName2 + \beta_2 VariableName3 + \beta_3 VariableName4 + \epsilon$$

Suppose we wanted to test the hypothesis that $\beta_2$ and $\beta_3$ were 0. Then, using our procedure, we would test this hypothesis as follows:

1. **Define** $H_0$.
   $H_0 : \beta_2 = 0$ and $\beta_3 = 0$.
2. **Define** $H_1$.
   $H_1 : \beta_2 \neq 0$ or $\beta_3 \neq 0$.
3. **Define** $\alpha$ = **Probability of Type I Error**.
   As usual, we set $\alpha = 0.05$.
4. **Define the test statistic and test.**
   We apply an Lagrange Multiplier test using the fact that we apply two restrictions in our

null hypothesis, then $n \cdot R^2$ from the regression of the residuals estimated from the restricted model on all the covariates from the unrestricted model is $\chi^2$ with two degrees of freedom.

Then, assuming that our null hypothesis restrictions are true:

$$T = n \cdot R^2 \sim \chi^2_2$$

5. **Define a rejection region given the test.**
We are in an Lagrange Multiplier test, which produces a test statistic from a one-sided distribution and $\alpha$ has been set as 0.05. We need to find some value of $C$ such that $P(T \geq C|H_0) = \alpha$, or the value $C$ such that the probability of observing some test statistic $T$ greater than $C$ assuming the null is true is $\alpha$. To find this value, we type in:

```
display invchi2(2, .95)
```

We find that $C = 5.9915$

Our rejection region is:

$$RR = (5.9915, \infty)$$

6. **Calculate and report the test statistic.** We first estimate the 'restricted' model, or the model implied by the null hypothesis and pull out residuals:

Figure 2: First Regression for LM Test

```
. regress VariableName1 VariableName2
```

| Source | SS | df | MS | Number of obs | = | 5,196 |
|---|---|---|---|---|---|---|
| | | | | F(1, 5194) | = | 48.23 |
| Model | 3.3455e+10 | 1 | 3.3455e+10 | Prob > F | = | 0.0000 |
| Residual | 3.6027e+12 | 5,194 | 693626695 | R-squared | = | 0.0092 |
| | | | | Adj R-squared | = | 0.0090 |
| Total | 3.6362e+12 | 5,195 | 699933032 | Root MSE | = | 26337 |

| VariableName1 | Coef. | Std. Err. | t | P>\|t\| | [95% Conf. | Interval] |
|---|---|---|---|---|---|---|
| VariableName2 | -2522.997 | 363.2861 | -6.94 | 0.000 | -3235.191 | -1810.803 |
| _cons | 38066.89 | 820.1431 | 46.41 | 0.000 | 36459.06 | 39674.71 |

```
. predict residuals, resid
(2,226 missing values generated)
```

We then regress the residuals on all covariates from the 'unrestricted model.'

Note that we get an $R^2$ value of 0.127. Multiplying this value by the number of observations, we get a test statistic $T = 654.69$.

Figure 3: Second Regression for LM Test

```
. regress residuals VariableName2 VariableName3 VariableName4
```

| Source | SS | df | MS | Number of obs | = | 5,151 |
|---|---|---|---|---|---|---|
| | | | | F(3, 5147) | = | 249.74 |
| Model | 4.5502e+11 | 3 | 1.5167e+11 | Prob > F | = | 0.0000 |
| Residual | 3.1258e+12 | 5,147 | 607314930 | R-squared | = | 0.1271 |
| | | | | Adj R-squared | = | 0.1266 |
| Total | 3.5809e+12 | 5,150 | 695314161 | Root MSE | = | 24644 |

| residuals | Coef. | Std. Err. | t | P>\|t\| | [95% Conf. | Interval] |
|---|---|---|---|---|---|---|
| VariableName2 | 527.6777 | 342.0477 | 1.54 | 0.123 | -142.8812 | 1198.237 |
| VariableName3 | 3058.401 | 122.7115 | 24.92 | 0.000 | 2817.835 | 3298.968 |
| VariableName4 | -10442.33 | 696.0757 | -15.00 | 0.000 | -11806.93 | -9077.724 |
| _cons | -28229.79 | 2041.704 | -13.83 | 0.000 | -32232.4 | -24227.18 |

7. **Do inference**. Under our choice of $\alpha = 0.05$, we see that the test statistic is very far into the rejection region. Therefore, we reject the null hypothesis at the 0.05 confidence level.

8. **Report p-values**. The p-value is $P(X > T|H_0)$, or the probability of observing an outcome as great or greater than the test statistic we observe given that the null is true. Note here that the p-value is reported under `Prob>F`. We see there that the p-value for this test listed in the table is essentially 0. We could derive this result by hand by typing in:

```
display chi2tail(2, 654.69)
```

Note that these results are similar to what we would get if we used an F test, as we also get a p-value of about 0 in that case. For observed test statistics that are closed to the rejection region boundary, there will in general be slight differences in p-values between the tests even though both tests are a means of doing inference on coefficient values.

## Prediction Using an 'Elegant' Method

Previously, we predicted values of the dependent variable using the `margins` command for specific predictions given certain values of covariates, and using the `predict` command for observations in our data. In class, the professor went over an 'elegant' method of prediction where we subtract off the values of the covariates at which we predict the expected mean and then run the regression normally. It's worth taking a minute to think about how this technique works. The 'true' model is:

$$VariableName1 = \beta_0 + \beta_1 VariableName2 + \beta_2 VariableName3 + \epsilon$$

Say we wanted to predict the conditional mean of $VariableName1$ at $VariableName2 = 3$ and $VariableName3 = 1$. Then, adjusting our 'true' equation, we have:

$$VariableName1 = (\beta_0 + 3\beta_1 + \beta_2) + \beta_1(VariableName2 - 3) + \beta_2(VariableName3 - 1) + \epsilon$$

Assuming MLR1-4 hold, then if we estimate a model of the following form:

$$VariableName1 = \hat{\gamma}_0 + \hat{\gamma}_1(VariableName2 - 3) + \hat{\gamma}_2(VariableName3 - 1) + \epsilon$$

Then $\hat{\gamma}_0$ will be consistent and unbiased for $\beta_0 + 3\beta_1 + \beta_2$, which is by construction the value of $\mathbb{E}[y|VariableName2 = 3, VariableName3 = 1]$. Furthermore, the standard error on $\hat{\gamma}_0$ will be the standard error on our prediction, which should be nearly identical to the standard error that we get from the `margins` command.

Thus, to implement this process, we would simply type in:

```
generate VariableName2new = VariableName2-3

generate VariableName3new = VariableName3-1

regress VariableName1 VariableName2new VariableName3new
```

And then look at the coefficient on the constant for both the estimated value (our prediction of $VariableName1$ at the relevant values of the covariates) and the standard error of the prediction.

## Standard Errors vs. Forecast Errors

Suppose that the 'true' model is:

$$VariableName1 = \beta_0 + \beta_1 VariableName2 + \epsilon$$

When estimating $\widehat{\mathbb{E}}[VariableName1|VariableName2]$, the standard errors that we get from either the previous 'elegant method' or the `margins` command are what we have learned are standard errors for predicted means, that is they are a standard error of the form:

$$\begin{aligned} se(\widehat{\mathbb{E}}[VariableName1|VariableName2...]) =& se(\hat{y}|VariableName2) \\ =& se(\hat{\beta}_0 + \hat{\beta}_1 VariableName2|VariableName2) \end{aligned}$$

Professor Smith drew a contrast between these form of standard errors, which are simply a standard error for a conditional mean, and the 'forecast' standard errors, which will always be larger. The standard errors in $se(\hat{y})$ come from the difference between $\hat{\beta}$ and $\beta$. If $\hat{\beta} = \beta$, then there would be no standard error for a conditional mean as there would be no variability in the estimates of $\beta$. Now, think about what happens when predicting a particular value of the dependent variable $VariableName1$. Then, since we are predicting $VariableName1$, there will always be an error in our prediction of $VariableName1$ since there is error in $VariableName1$ that is unexplained by $VariableName2$. Therefore, even if $\hat{\beta} = \beta$, there will still exist variation in the 'true' value of $VariableName1$ that will be uncaptured by our estimate. Thus, the forecast error variance will always be greater than 0.

Now, the true value of $VariableName1 = \beta_0 + \beta_1 VariableName2 + \epsilon$. Therefore:

$$\begin{aligned} V(\widehat{\mathbb{E}}[VariableName1|VariableName2] - VariableName1|VariableName2) = \\ V(\hat{y}|VariableName2) + V(\epsilon|VariableName2) = V(\hat{y}|VariableName2) + \sigma^2 \end{aligned}$$

Therefore, if we are considering our estimates as direct predictions for specific individuals and not as a conditional mean, then the standard deviation of our estimate is:

$$\begin{aligned} sd(\widehat{VariableName1}^{forecast} - VariableName1|VariableName2) = \\ \sqrt{V(\widehat{VariableName1}|VariableName2) + \sigma^2} \end{aligned}$$

Hence, the standard error is:

$$\begin{aligned} se(\widehat{VariableName1}^{forecast} - VariableName1|VariableName2) = \\ \sqrt{\hat{V}(\widehat{VariableName1}|VariableName2) + \hat{\sigma}^2} \end{aligned}$$

Thus, we can calculate the standard error for a forecast by squaring the standard error from a prediction, adding the estimated variance of the error term and taking a square root.

We can calculate standard errors for forecasts for all observations in our data by typing after a regression:

```
regress VariableName1
```

```
predict forecaststandarderror, stdf
```

We can calculate standard errors for the predicted mean for all observations in our data by typing:

```
predict forecaststandarderror, stdp
```

We can calculate standard errors for forecasts at the level of $VariableName2 = 5$ by typing:

```
adjust VariableName2=5, stdf
```

And we can finally calculate standard errors for the predicted mean for the level of $VariableName2 = 5$ by:

```
adjust VariableName2=5, stdf
```

## Categorical Variables in Models

Floating in the background of our previous discussion on interacted variables is a specific way of thinking about categorical variables that is more common in economics than in other disciplines. Consider first a simple dummy variable $VariableName2$ that takes on values 1 and 0:

$$VariableName1 = \beta_0 + \beta_1 VariableName2 + \epsilon$$

So, for observations that have $VariableName2 = 0$, then $VariableName1 = \beta_0 + \epsilon$, and for observations that have $VariableName2 = 1$ then $VariableName1 = \beta_0 + \beta_1 + \epsilon$.

Therefore:

$$\begin{aligned}
\beta_0 &= \mathbb{E}[VariableName1|VariableName2 = 0] \\
\beta_1 + \beta_0 &= \mathbb{E}[VariableName1|VariableName2 = 1] \\
\beta_1 &= \mathbb{E}[VariableName1|VariableName2 = 1] - \mathbb{E}[VariableName1|VariableName2 = 0]
\end{aligned}$$

Thus, categorical dummy variables like $\beta_1$ (if we have not included interaction terms) can be interpreted as the difference in the expected value of the dependent variable conditional on all other characteristics being held constant. Let's say we were looking at earnings as a dependent variable, and our single dummy variable was a variable for sex. Then, $\beta_0$ would be the expected value of earnings among women, and $\beta_1$ would be the expected value of earnings among men minus the

expected value of earnings among women. Thus, $\hat{\beta}_0$ would be the mean of earnings among women in our survey, and $\hat{\beta}_1$ would be the mean of earnings for men minus the mean of earnings for women.

Note that even though we wanted to include categories describing sex, we only included a single dummy variable defining sex instead of two. This feature of setting up these equations is consistent for any number of categories. If we're including dummy variables describing categories, we always exclude one category. So, if we have dummy variables describing race in six categories and if we want to run a regression where we include dummy variables for these race variables, we would only include five categories as dummy variables as including the sixth category would make the sum of all categories collinear with the constant term. The 'omitted category,' however, is still implicitly present. As in the case above with only two categories and no other covariates, the constant term is the expected value of the dependent variable for the omitted category.

Now let's consider creating variables coding these different categories. Let's say that we have a categorical variable $Variable2$ that takes on 4 values, and we would like to include include it in our regression. We could manually go in and construct the dummy variables as follows:

```
generate Variable2category1=.

generate Variable2category2=.

generate Variable2category3=.

generate Variable2category4=.

replace Variable2category1=1 if Variable2==1

replace Variable2category2=1 if Variable2==2

replace Variable2category3=1 if Variable2==3

replace Variable2category4=1 if Variable2==4

replace Variable2category1=0 if Variable2==2 | Variable2==3 | Variable2==4

replace Variable2category2=0 if Variable2==1 | Variable2==3 | Variable2==4

replace Variable2category3=0 if Variable2==1 | Variable2==2 | Variable2==4

replace Variable2category4=0 if Variable2==1 | Variable2==2 | Variable2==3
```

A much faster way of generating categorical variables, however, is to use `tabulate ... , generate(...)` We could've performed the same operation as above and created the exact same variable categories if we had just written:

```
tabulate Variable2, generate(Variable2category)
```

Then, STATA generates the same variable names as above and creates a dummy variable for

each category. Then, we could run a regression including these categories by writing:

```
regress VariableName1 Variable2category2 Variablte2category3 Variable3category4
```

We've written our category variables with numbers after them because we can write the same regression a little more compactly. We can use a dash between variable names to tell STATA to use all variables that occur between these variables on either side of the dash. So, since these variables were created consecutively, we could tell STATA to run the exact same regression using:

```
regress VariableName1 Variable2category2 - Variable3category4
```

Note that we have chosen to exclude $Variable2category1$. Thus, the coefficient on $Variable2category2$ is an estimate of the difference between the expected value of $VariableName1$ for individuals that have $VariableName2 = 2$ and the expected value of $VariableName1$ for individuals that have $VariableName2 = 1$. As we talked about in class, the base level choice does not change the final predicted values of the dependent variable for different subgroups, but it does change the ease of interpretation of coefficients. The coefficients always measure a difference in conditional expected value of the dependent variable for the group defined by a specific dummy variable with the expected value of the dependent variable for the omitted category.

Now, we just ran this regression by creating new separate dummy variables. We could have run the same regression without directly having created the variables by typing the following command, which estimates the same regression without creating separate variables in the dataset. When running large regressions, sometimes this shortcut can be faster as it avoids creating a number of new variables.

```
regress VariableName1 i.VariableName2, baselevels
```

Note the inclusion of `baselevels` tells STATA to specify which level of the categorical variable $VariableName2$ is chosen as a base level. We can directly tell STATA to directly choose $VariableName2 = 2$ as a base level by typing:

```
regress VariableName1 ib2.VariableName2, baselevels
```

If we wanted to both run the regression and generate dummy variables, we could just add to this command:

```
xi:  regress VariableName1 i.VariableName2
```

## Inference with Categorical Variables in Models

Consider hypothesis testing in this framework. In the previous regression, we directly specified the covariates we included, so if we wanted to test whether the coefficient on $Variable2category3$ is different than 0 (which is, by the way, a test of the difference in means of $VariableName1$ between individuals in category 3 and individuals in the omitted category, or category 1) we could type:

```
regress VariableName1 Variable2category2 Variablte2category3 Variable2category4

test Variable2category3=0
```

If we use the shortcuts mentioned before, then we need to refer to variables somewhat differently. If we mention the separate dummy variables using the `i` prefix, then we would refer to the categorical dummy variable that equals 1 for observations where $VariableName2 = 3$ as `3.VariableName2`, and hence we would test the hypothesis that the coefficient on this categorical variable is 0 as follows:

```
regress VariableName1 ib2.VariableName2, baselevels

test 3.VariableName2=0
```

If we create the separate dummy variables using the *xi* prefix, then STATA has created additional dummy variables that are referenced in the regression output, and we need to refer to them by the titles used in the regression:

```
xi:  regress VariableName1 i.VariableName2

test _IVariableN_3=0
```

## Combining Different Categorical Variables Together

Consider a case where we have one variable $VariableName2$ that takes on values of 0 and 1, and another variable $VariableName2$ that takes on values of 0 and 1 Note that together these two different categorical variables define 4 different categories ($VariableName2 = 0$ and $VariableName3 = 0$ for one category, $VariableName2 = 1$ and $VariableName3 = 0$ for another category, etc.)

We could create these additional categories long hand by coding up dummy variables for each of the categories. Continuing with our previous example, we could do this as follows:

```
generate cat1 = .

generate cat2 = .

generate cat3 = .

generate cat4 = .

replace cat1=1 if VariableName2==0 & VariableName3==0

replace cat2=1 if VariableName2==1 & VariableName3==0

replace cat3=1 if VariableName2==0 & VariableName3==1
```

```
replace cat4=1 if VariableName2==1 & VariableName3==1

replace cat1=0 if (VariableName2==1 & VariableName3==0) | (VariableName2==0
& VariableName3==1) | (VariableName2==1 & VariableName3==1)

replace cat2=0 if (VariableName2==0 & VariableName3==0) | (VariableName2==0
& VariableName3==1) | (VariableName2==1 & VariableName3==1)

replace cat3=0 if (VariableName2==0 & VariableName3==0) | (VariableName2==1
& VariableName3==0) | (VariableName2==1 & VariableName3==1)

replace cat4=0 if (VariableName2==0 & VariableName3==0) | (VariableName2==1
& VariableName3==0) | (VariableName2==0 & VariableName3==1)
```

Now consider a model of the form:

$$VariableName1 = \beta_1 + \beta_2 cat2 + \beta_3 cat3 + \beta_4 cat4 + \epsilon$$

Then:

$$\begin{aligned}
\beta_1 =& \mathbb{E}[VariableName1 | VariableName2 == 0 \& VariableName3 == 0] \\
\beta_2 =& \mathbb{E}[VariableName1 | VariableName2 == 1 \& VariableName3 == 0] - \\
& \mathbb{E}[VariableName1 | VariableName2 == 0 \& VariableName3 == 0] \\
\beta_3 =& \mathbb{E}[VariableName1 | VariableName2 == 0 \& VariableName3 == 1] - \\
& \mathbb{E}[VariableName1 | VariableName2 == 0 \& VariableName3 == 0] \\
\beta_4 =& \mathbb{E}[VariableName1 | VariableName2 == 1 \& VariableName3 == 1] - \\
& \mathbb{E}[VariableName1 | VariableName2 == 0 \& VariableName3 == 0]
\end{aligned}$$

A somewhat faster way of accomplishing the same goal of allowing would be to create interaction terms:

```
generate Variable2interact = Variable2 * Variable3
```

Now consider a model of the form:

$$VariableName1 = \gamma_1 + \gamma_2 Variable2 + \gamma_3 Variable3 + \gamma_4 Variable2interact + \epsilon$$

Then:

$$
\begin{aligned}
\gamma_1 =& \mathbb{E}[VariableName1|VariableName2 == 0 \& VariableName3 == 0] \\
\gamma_2 =& \mathbb{E}[VariableName1|VariableName2 == 1 \& VariableName3 == 0] - \\
& \mathbb{E}[VariableName1|VariableName2 == 0 \& VariableName3 == 0] \\
\gamma_3 =& \mathbb{E}[VariableName1|VariableName2 == 0 \& VariableName3 == 1] - \\
& \mathbb{E}[VariableName1|VariableName2 == 0 \& VariableName3 == 0] \\
\gamma_4 =& \mathbb{E}[VariableName1|VariableName2 == 1 \& VariableName3 == 1] - \\
& \mathbb{E}[VariableName1|VariableName2 == 0 \& VariableName3 == 0] - \\
& (\mathbb{E}[VariableName1|VariableName2 == 1 \& VariableName3 == 0] - \\
& \mathbb{E}[VariableName1|VariableName2 == 0 \& VariableName3 == 0]) - \\
& (\mathbb{E}[VariableName1|VariableName2 == 0 \& VariableName3 == 1] - \\
& \mathbb{E}[VariableName1|VariableName2 == 0 \& VariableName3 == 0])
\end{aligned}
$$

Hence, $\beta_1 = \gamma_1$, $\beta_2 = \gamma_2$, $\beta_3 = \gamma_3$, but $\gamma_4 = \beta_4 - \beta_3 - \beta_2$.

Although most coefficients are the same, the interaction term has a slightly different interpretation here than the fourth subcategory variable. However, the final conditional mean results are the same for different subgroups when we estimate these models.

Now, consider generalizing these results. Suppose we have four categories for one variable and three categories for another variable. If we wanted to estimate a model

Thus, if we have four categories for one variable and three categories for another variable, then we could either generate category dummy variables for the subcategories or we could create dummy variables defining the category levels, and then fully interact all the dummy variables across categories.

Alternatively, instead of directly interacting the dummy variables in STATA, we could accomplish the goal by interacting the category variables as follows with an `i.` prefix, and a `##` interaction term, which creates dummy categories for the categories separately and includes the individual dummy variables and the interacted dummy variables.

```
regress VariableName1 i.VariableName2##i.VariableName3
```

As previously, when using these categorical variable shortcuts, we can directly specify the omitted category after `i.b`. So, if we want to specify the subgroup where both $VariableName2 = 3$ and $VariableName3 = 1$ as the omitted category, and tell STATA to report which category were chosen using the `baselevels` suffix:

```
regress VariableName1 ib3.VariableName2##ib1.VariableName3, baselevels
```

### Inference with Combining Different Categorical Variables Together

If we use the previous shortcuts for interactions, then we refer to the estimated coefficients in postestimation hypothesis tests in a way that is similar to our previous method. If we estimate the following regression:

```
regress VariableName1 ib1.VariableName2##ib1.VariableName3, baselevels
```

Then we can test the null hypothesis that the coefficient on the dummy variable for $VariableName2 = 3$ is 0 as follows:

```
test 3.VariableName2=0
```

We test the null hypothesis that the coefficient on the dummy variable for $VariableName3 = 2$ is 0 similarly:

```
test 2.VariableName3=0
```

And we finally test the null hypothesis that the coefficient on the dummy variable for the the interaction of the dummy variables for $VariableName2 = 3$ is 0 and $VariableName3 = 2$ is 0 as:

```
test 3.VariableName2#2.VariableName3=0
```

## The `xtile` Function

Previously, we used the `egen` command to compute percentiles and specify whether or not observations had values that were in higher and lower percentiles. Sometimes, we to create variables specifying percentiles more directly. So, for example, we might want to sort observations into 5 quantiles of $VariableName1$ and create a new variable titled $VariablePercentile$ that records the relevant quantile. To do this, we could type:

```
xtile VariablePercentile = VariableName1, nquantiles(10)
```