**Stat 401 – extra information on Lab 11 topics**

**Contents:**

Creating indicator variables using a formula

More on JMP's coding of indicator variables

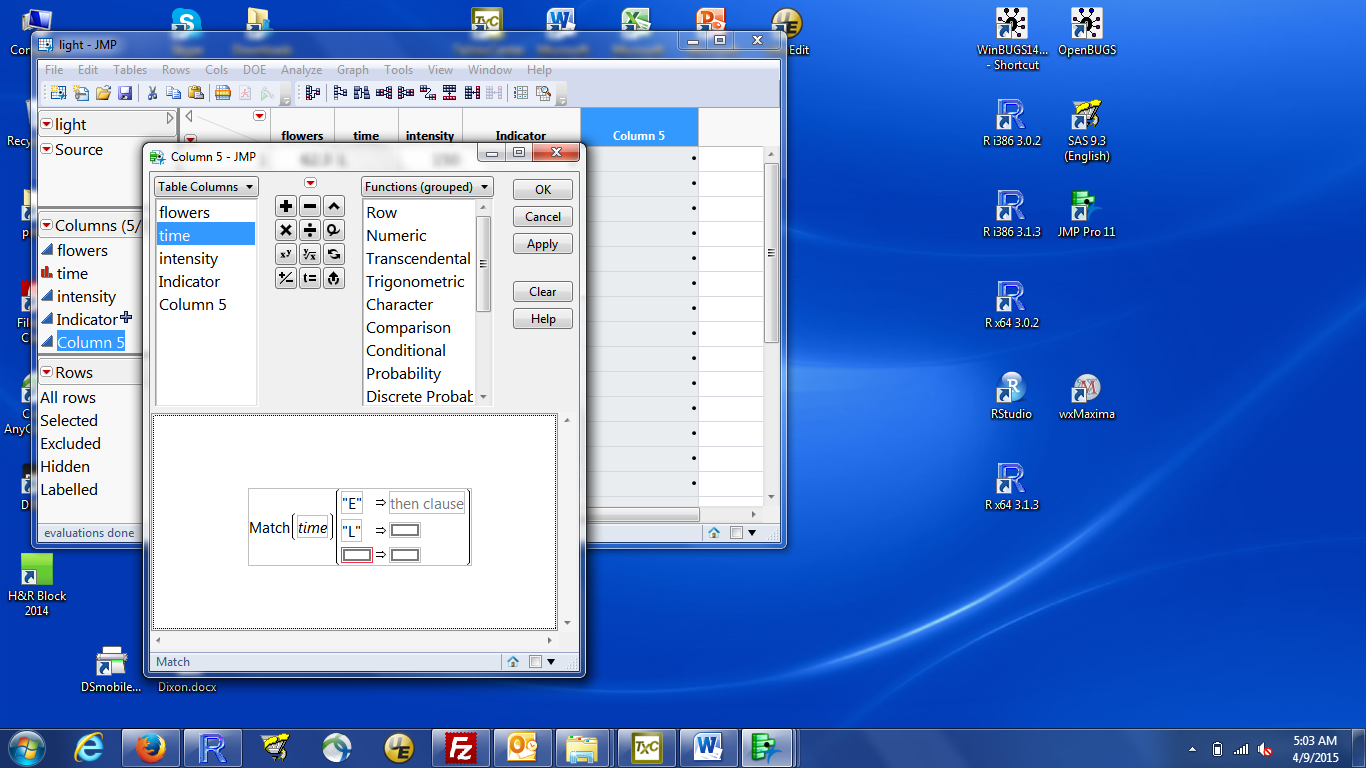
Model comparison F tests without hand-computation

**Creating indicator variables using a formula:**

The formula function that best converts codes to values of an indicator variable is match(). JMP provides a ‘fill in values automatically’ option, which is extremely helpful but hard to find. The JMP 13 documention says it appears automatically. It only does if you do the operations in the correct order. Here's what works:

Create a new column one of the usual ways (e.g., double click on a blank column, select column properties then formula). The usual formula construction box appears. Right click on the variable that defines the groups (e.g. time in the light / flower production example). You should see time go into the formula box.

Right click Conditional, then select match. You should see the following template with some values partially filled in:

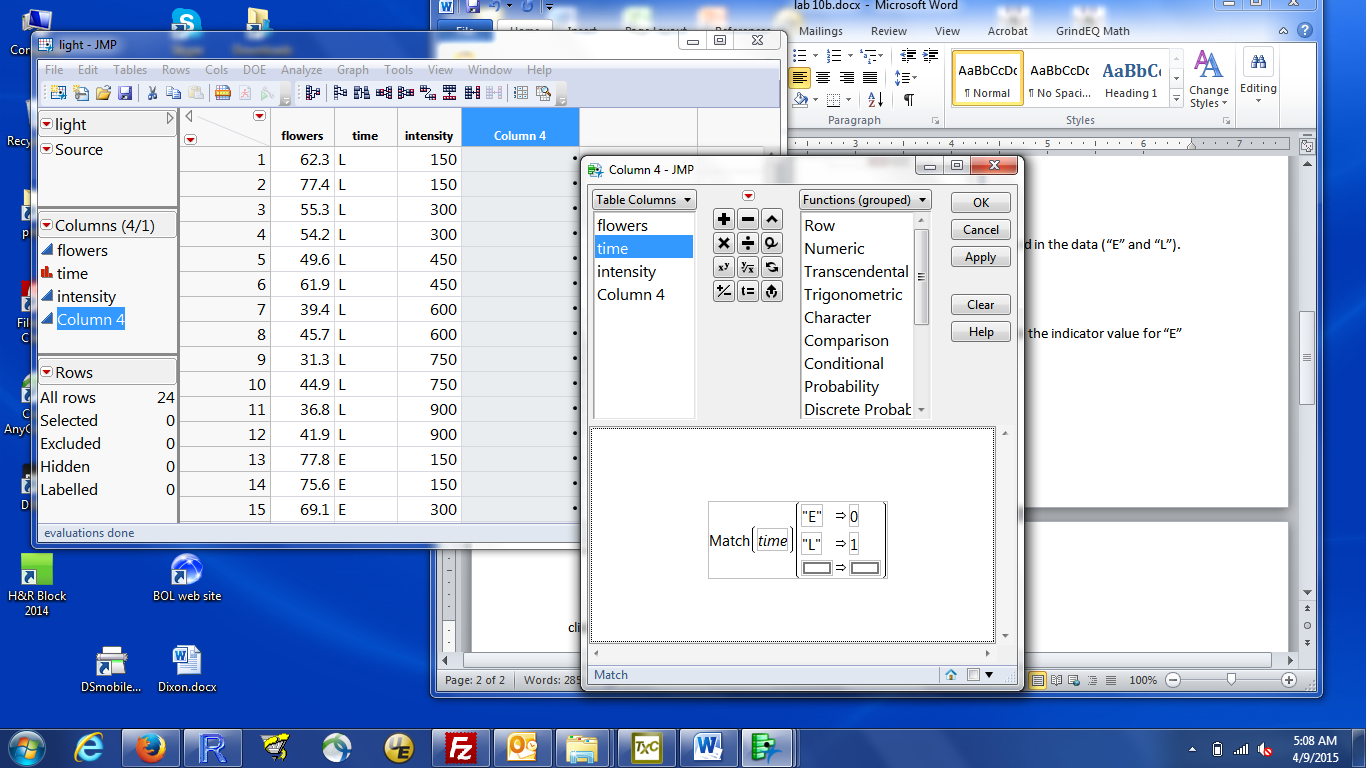


JMP has filled in the name of the defining variable (time) and the levels found in the data (“E” and “L”). The third row is for ‘not in the above list’, which we won’t use.

left-click the box to the right of “E” => (with the text “then clause”) and fill in the indicator value for “E” groups. To match the class example, this should be 0.

repeat for “L”, but the indicator should be a 1 for the “L” observations.

This is what that dialog looks like when filled in:



click OK. A new column will be added. You can change the column name to something more interesting in any of the usual ways (double left click on the name, or right click on the name/column, select Column info, then change the name, or select the column, then Cols / Column Info).

**More on JMP's coding of indicator variables**

You can see the indicator variable that JMP constructs after fitting the model by:

left click the red triangle by Response flowers (top left of the results box)

select Save Columns / Save Coding Table

You get a new data window with a new variable time[E]. This is the indicator JMP created.

If your nominal variable has *k* levels, you will get *k-1* new indicator variables.

If you look at the coding table, you will see that the JMP-created indicator has values of +1 (for E) and -1 (for L)

The parameter estimates include a Term labeled time[E]. This is the regression estimate for the indicator variable with +1/-1 coding. This is shown in the Parameter Estimates box in the screen shot below.

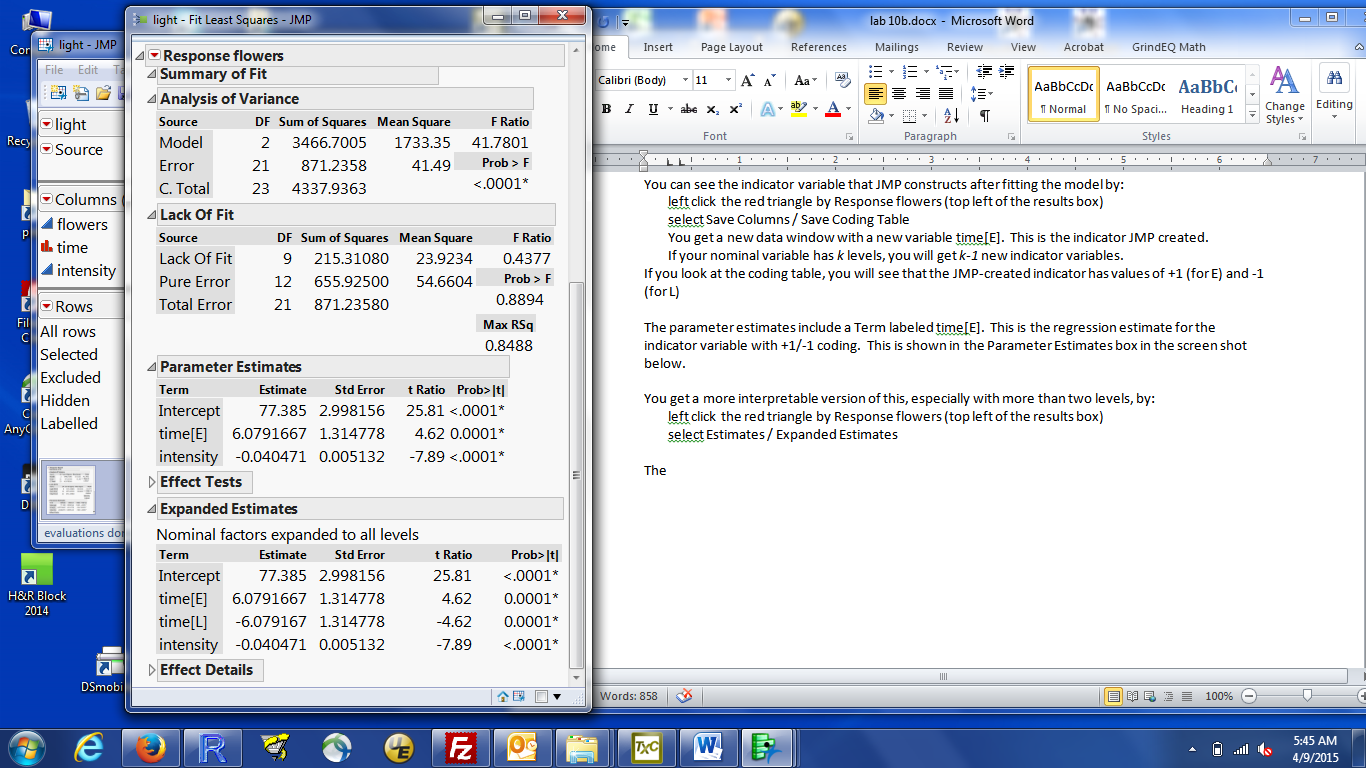
You get a more interpretable version of this, especially with more than two levels, by:

left click the red triangle by Response flowers (top left of the results box)

select Estimates / Expanded Estimates

The Expanded Estimates box has one value for each level of the nominal variable. This is the amount that is added to the intercept when an observation is in the specified group.

Here are both the Parameter Estimates and Expanded Estimates output:



The intercept for an “E” observation is Intercept + time[E] = 77.395 + 6.079 = 83.46 ; that for an “L” observation is Intercept + time[L] = 77.395 + (-6.079) = 71.30.

The regression coefficients (e.g. the 6.079) changes for different choices of indicator coding. It’s 12.158 for 0/1 coding with last = 0. But, no matter what coding is used, the intercept for the E group is always 83.46; that for the L group is always 71.30.

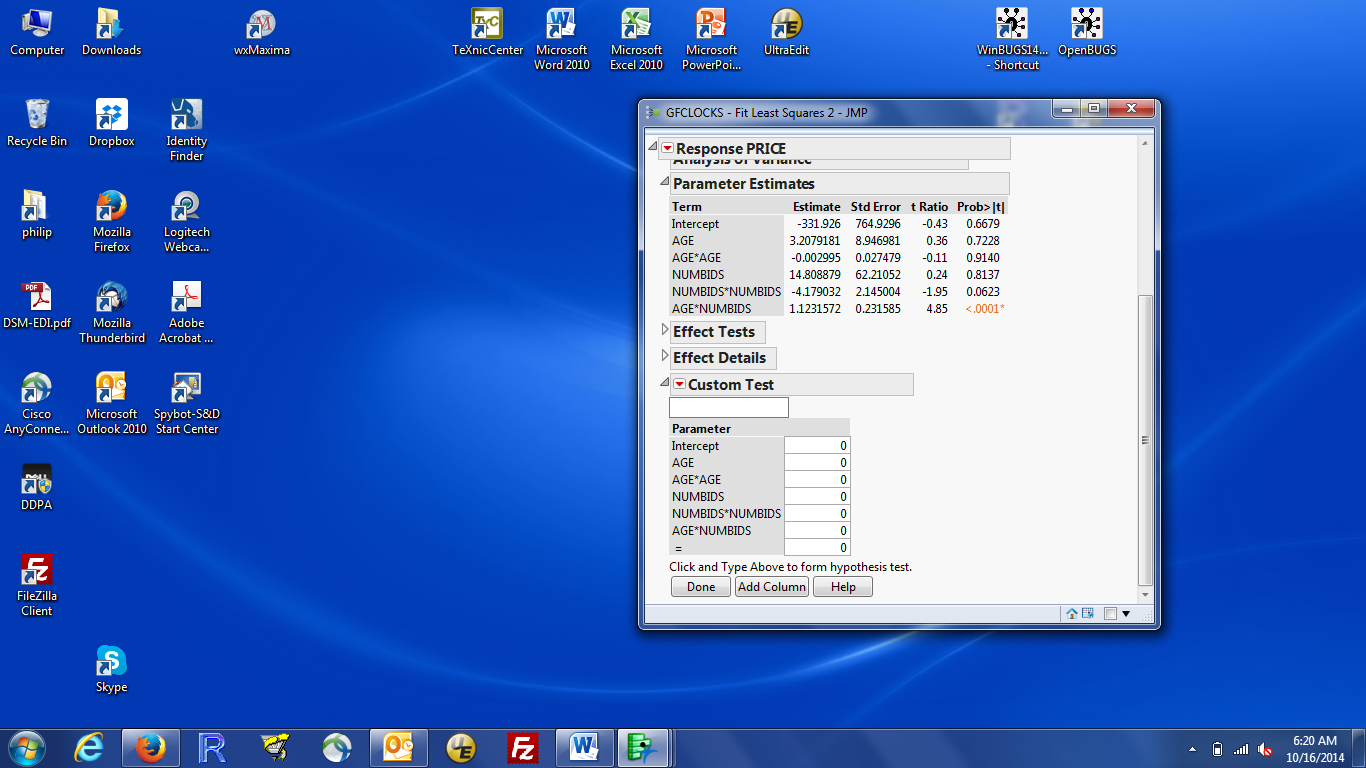
**Model comparison F test for two nested models:** JMP gives you the “all variables” vs “only the intercept” model comparison automatically (the Analysis of Variance table in the JMP output). Other model comparisons can be obtained in either of two ways.

1. Fit each model separately (two runs of Analyze / Fit Model). Look at the Analysis of Variance result for each model, extract the Error DF and Sums of Squares. Calculate the F statistic by hand. The p-value has to be obtained from printed tables.
2. JMP can test arbitrarily complex null hypotheses about parameters. This includes the null hypotheses that correspond to model comparisons. My example will test whether both quadratic terms are needed in a model for the Grandfather clocks auction price:

i.e., the null hypothesis that and

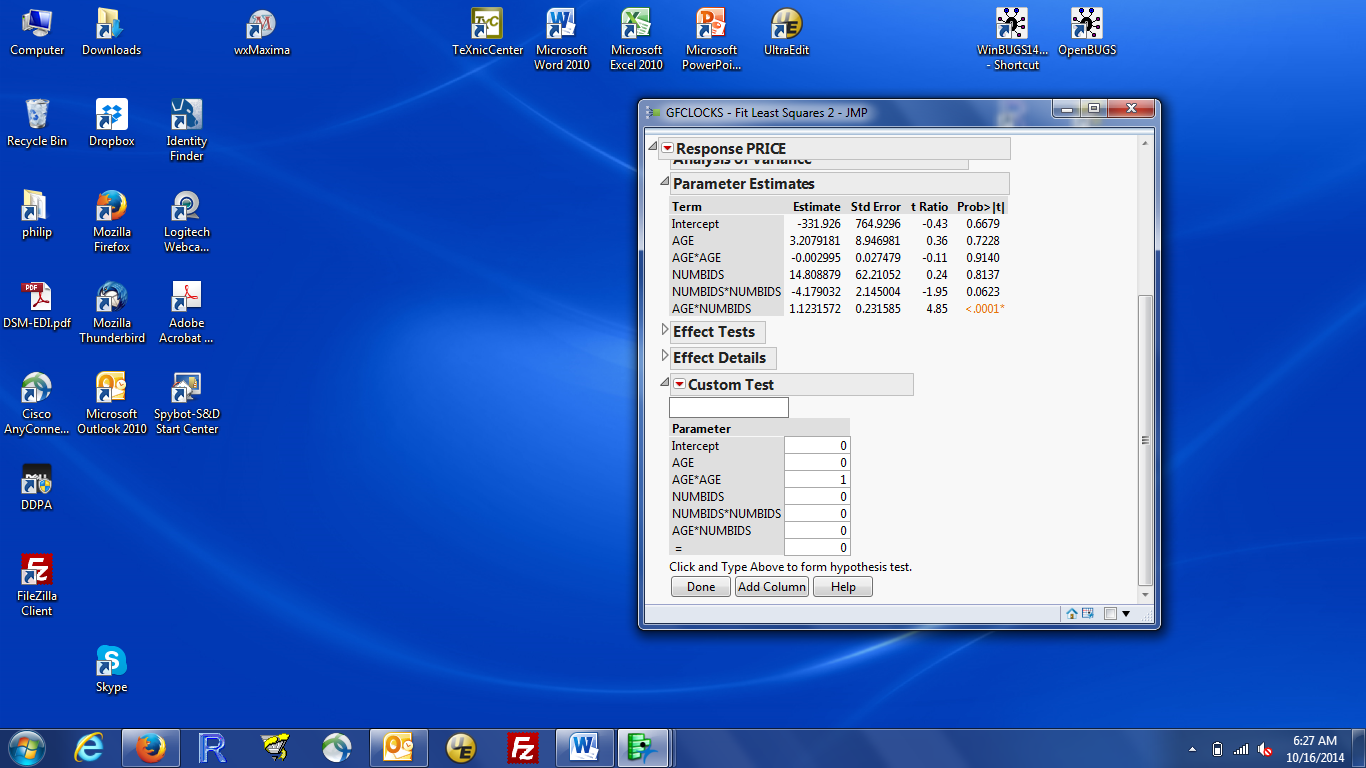
The data are in gfclocks.txt.

Fit the **full** model (the 6 parameter model), then click the red triangle by Response PRICE, select Estimates / Custom Test. You should get a dialog box looking like:

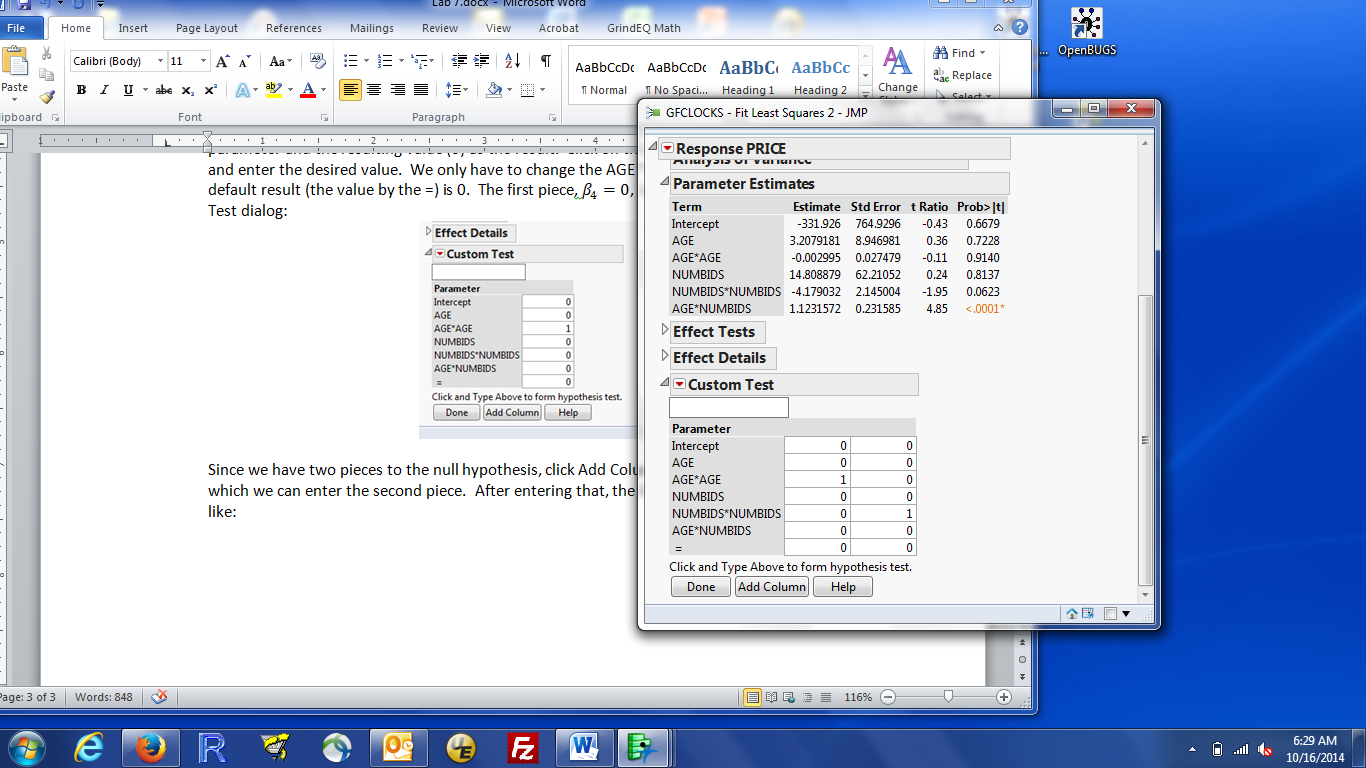


This allows you to specify each component of the **Null** hypothesis (here, and ). To see how to specify this to JMP, remember that the null hypothesis we want to test is exactly the same as: and . There are two pieces to this null hypothesis. The first concerns β4; the second concerns β5. We have to specify each piece.

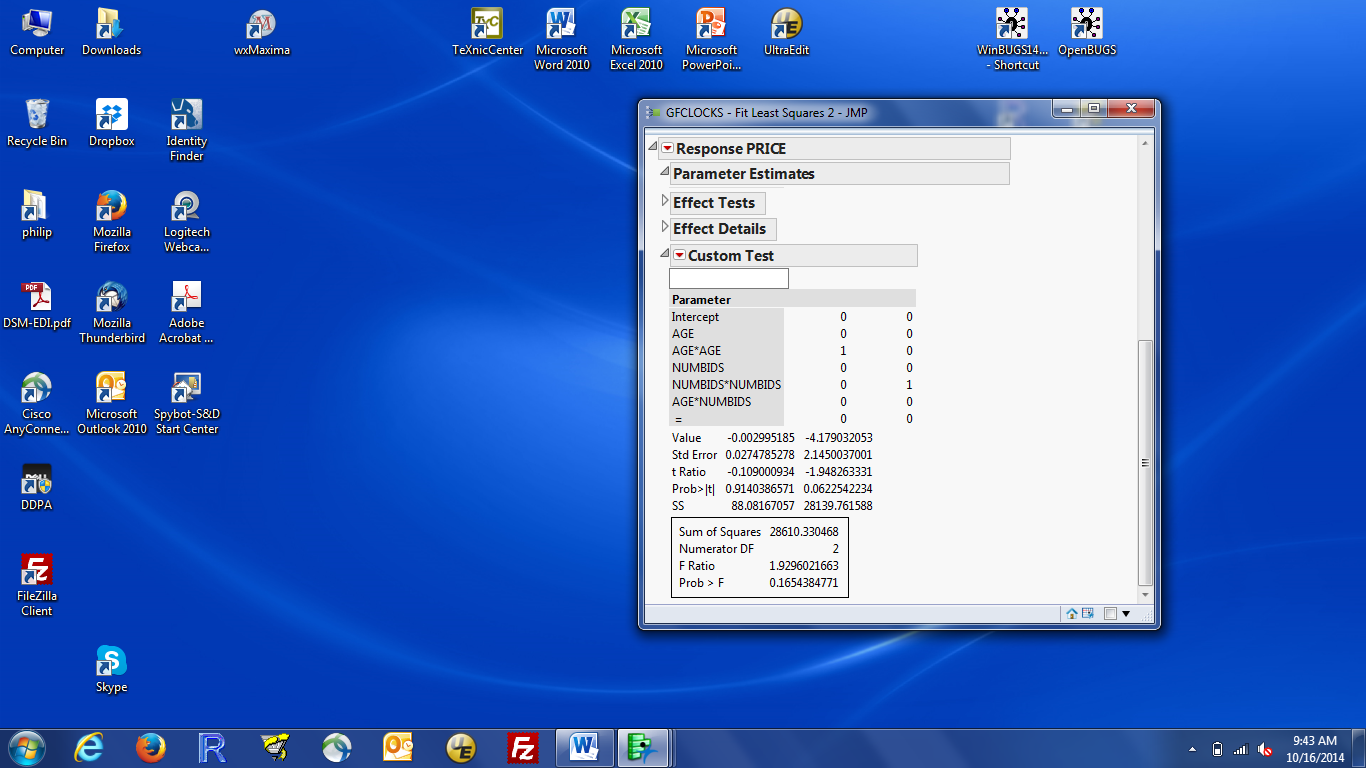
Each piece is specified to JMP by entering the coefficient (1) adjacent to the appropriate parameter and the resulting value (0) as the result. Click on the number next to each parameter and enter the desired value. We only have to change the AGE\*AGE coefficient because the default result (the value by the =) is 0. The first piece, , is the following in the Custom Test dialog:



Since we have two pieces to the null hypothesis, click Add Column to get a second column in which we can enter the second piece. When done, the Custom Test dialog should look like:



Click Done to run the test. The output is on the next page.



The two columns are information about each piece separately. In this case (since the coefficients are 1 for each piece), that is the same information available from the parameter estimates box. The output we want is in the box at the bottom of the window. Sum of Squares is the change in SS Error between the two models; Numerator DF is the change in DF Error between the two models. JMP goes directly to the F statistic and gives you the p-value. Here, my conclusion would be no evidence of a quadratic relationship for either age or numbids. Practically, I would then omit those two variables from my model.

**Note of caution:** JMP does exactly what you tell it to do. It can’t read your mind. If you put a 1 in the wrong place, you will get the wrong results because JMP is testing the wrong hypothesis. In particular, it is easy to put two 1’s in the same column, instead of in two different columns. If you put two 1’s in the same column, you are asking JMP to test the hypothesis:

. Very different!

If you aren’t sure that you’ve specified your test correctly, I suggest two checks:

1)Does the test have the correct d.f.? If there are two pieces in your null hypothesis, the test should have numerator df = 2. Alternatively, the numerator df should equal the number of equal signs in your null hypothesis.

2) Does the test have the correct SS? Check by fitting the two models you want (so the models being compared are clear), then hand computing the change in SS. If that’s correct, the rest of the output is almost certainly correct.