EngineRoom's General Linear Modeling tool is located in the “Analyze” menu under “Regression Analysis…”

GLM is an extension of the multiple regression tool in EngineRoom, and in this video we will go over the main features GLM has that make it a more advanced regression analysis tool.

First we will create a mixed effect model with the tool. Here we will be using our “glm_random_block_example” dataset where we have a raw material batch, two process input variables (Temperature and Pressure), and a response output (Strength). We will start the analysis by dragging “Temperature” and “Pressure” into the Continuous Variables Dropzone. Then drag “Strength” over to the Response Variable Dropzone to open the “Basic Setup” menu. Before adding in our batch variable, let’s create a basic least squares model using the default settings.

Notice that if we just use our input variables Temperature and Pressure in our model, we don't get a very powerful model. We have a low R^2 of 45% and Pressure was removed from the model for not being statistically significant. What we will soon see is that we have too much error (or noise) in the model that we can reduce using a random block.

Now let’s go back to the beginning by selecting “Edit Options” and let’s add “Raw Material Batch” by dragging it into the Categorical Variables Dropzone. This process has a lot of random batch to batch variation in strength, and a new batch is brought in every 5 data points. Since the batches in this dataset will never be used again but we want to account for the variation they contribute to our response, we will set the variable to “Random” instead of “Fixed” by hitting the slider.

We can now select “Calculate” again to see our results. Here we see a much improved model with an R^2 of 93% and both our fixed effects in the model. We didn't change any of the data from the last model, we just associated some of the error in the model to a known source, allowing us to have better power to see significant effects in our model. If we look at the Variance Components table, we can see that 82% of the error left over from our model can be attributed to batch to batch variation.

Now we will close out to start our second example where we will use higher order terms in our model. We will again open GLM and this time use the “glm_interactions_example” where we will use the Weight and Activity independent variables to predict Bone Density.

Drag Weight and Activity over to the Continuous Variables Dropzone and then Bone Density to the Response Variable Dropzone. Leaving the basic setup the same, we will go to “Advanced Setup” this time. Here we will select “Edit” next to “Interactions and Higher Order Terms”. We are going to add both the quadratics and the interaction terms to our model by selecting Activity and Weight on the left and selecting “Add Higher Order Terms” and “Add Interaction Terms”. Once those are added we can continue to generate our model by selecting “Calaculate”.

We will come to familiar output where we see that only the weight quadratic term was removed from the model. The R^2 of this model is not very high at 53%, but we expected a large amount of noise with this response so we are satisfied with our higher order model that helps us understand how our two factors influence Bone Density. Let's close out of this analysis to start our final example utilizing nested variables.

In this example we will be using the "glm_nested_example" dataset where a bakery has run a designed experiment to see what temperature and time they should pull their bread out of each rack of each oven to achieve the optimal bread weight of 900 grams. They have two ovens that each have a top and bottom rack. When we fit our model, we will need to nest Rack in Oven because the rack selection is reliant on what oven selection you make.

In GLM we will drag Oven and Rack over to the Categorical Variables Dropzone and Temperature and Time over to the Continuous Variables Dropzone. Then we will drag Weight to our Response Variable Dropzone. Leaving the basic setup the same we will go to "Advanced Setup" and select "Edit" next to "Nested Variables".

Here we will specify that "Rack" is nested in "Oven" and then select "Calculate". In our output we see a good fitting model and since we have categoric variables in our model, we have new tables that test the average difference between each unique level of our categoric variables. Notice how Rack has 4 unique levels since it is nested in Oven. We can also visualize these comparisons by selecting the "confidence intervals" tab in our visual output pane.