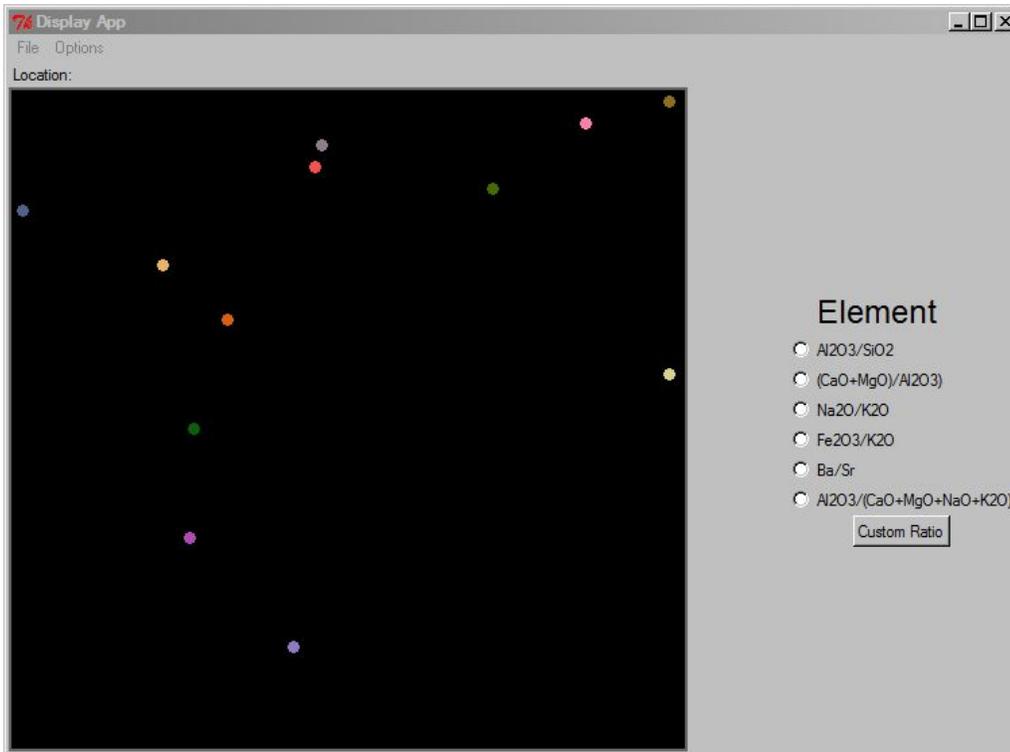


# GUI Development

Our goal for the GUI this week was primarily to integrate our DataSet class into the rudimentary display that Professor Maxwell provided us last week. Concretely, we wanted to be able to select an element or ratio and the Excel file, and have the appropriate data display as depth vs. concentration.



This is the current GUI. The points are from the primary data set. The vertical axis is depth (0cm below ash at the top, 70cm below ash at the bottom), and the horizontal axis is concentration of Al<sub>2</sub>O<sub>3</sub> (about 16% on the left, and 20% on the right). No radio button is selected because the selection was made using "custom ratio," entering "Al<sub>2</sub>O<sub>3</sub>."

## Element Selection

Cassi only wants to plot the concentrations of a single element or ratio at a time. Radiobuttons are appropriate for this requirement. The options should include all metal oxides and trace elements in the file in addition to the common ratios. Since it is probable that a new ratio will some day be of interest, an option to create a custom ratio is also desirable. Different analyses include different elements, so it makes sense to create the options after a file is selected. At this stage of the development, we have not implemented this functionality, so the user can only either select one of the common ratios, or click the "custom ratio" button. The button generates a dialog window that asks for an expression. Anything from a single compound name to a complicated equation with parentheses, scalars, elements, and common operators can be processed correctly (we created a module named 'expressions' that can evaluate valid equations; invalid equations currently cause errors). Element selection does not make any immediate visible change, but it does set the value of a StringVar, which is used upon data selection.

## Data Selection

This is very straightforward. Our DataSet class can be constructed by passing it a file name, so we adjusted the display's handleOpen method so that the return type from the file picker dialog is the file name (as a String) instead of a file object, like it was originally. The program stores the DataSet, and gets the data to plot by calling computeEquation(), passing in the value of the StringVar. The data is still in the data space, so we made a transform method to send it through a "pipeline." Right now, it simply scales and translates it to fill the plotting canvas on the screen, but we can just alter that method to improve the logic. After being transformed, the data easily be plotted by drawing an oval at each pair of coordinates.

## Areas to Improve

- once a data set has been chosen, add radio buttons for every element
- consider moving radiobutton canvas onto a dialog box that only appears if requested

- add labeled, scaled axes
- at the top, there is a label "Location: "; extract the location from the meta data and display it
- implement error-checking for custom ratios
- make a more intellegent, elegant pipeline
- allow user to adjust scales
- create side-by-side plots