Using RegEx in Expressions

Introduction
Using regular expressions in Designer isn't just limited to the RegEx tool. Functions involving regular expressions can also be applied in tools that contain expression editors, such as the Formula tool. Using RegEx in expressions can be a powerful way to augment the scope and application of regular expressions in data analysis.

Data
A dataset contains information on loans issued by various global financial institutions. Each data entry contains information on the start and end date of the loan, the alphanumeric loan identifier, the two letter country code and country name to which the loan was issued, the guarantor of the loan, and the principal amount of the loan. To better delineate individual values and evaluate their completeness, apply functions that support the use of regular expressions.

Drag a Formula tool onto the Canvas and connect it to the Input Data tool.

Replace Characters
One particularly difficult aspect of analyzing this dataset is that the end date of the loan and the loan identifier are not separated by any kind of delimiter. Create an expression in a new column called "Replace" to insert a pipe into each data entry to delimit these values.

To do so, we'll use a function that supports the use of regular expressions. These functions are found in the Expression Editor's function library for String data types. To perform the task at hand, inserting a pipe delimiter into our data, we'll use the RegEx Replace function. We'll click this function to insert it into our expression editor.

Open the function library and expand the String functions. Then, click "RegEx Replace" to insert the function into the Expression Editor.

The RegEx Replace function requires three parameters: the string column to modify, the pattern to identify in the string column and the replacement text to insert when the pattern is found.

First, select the string column to modify. Use the Columns and Constants button to insert "Column1" into the expression.

Now, create the regular expression to identify a pattern. Note that the Expression Editor does not contain a RegEx library, meaning regular expressions must be typed manually.

Identify the pattern in the data that represents the start and end dates of the loan. Then, enclose the regular expression in quotes.

The data to identify in the column [Column 1] is the loan's start and end dates, which are represented by the 4 digit year, hyphen, two digit month, hyphen, 2 digit day. The start and end dates are separated by a comma. It is at this point in the cell that a pipe should be inserted.

When the regular expression identifies the specified pattern, insert a pipe delimiter. In the expression, insert a pipe, surrounded by double quotes, as the third parameter. Then, click Submit.

The Data Preview in the Formula tool's configuration provides a view of the output from the current expression. Currently, the start and end dates of the loan have been completely replaced by the replacement text: the pipe delimiter. In some cases, a full replace might be the objective. However, to preserve the data representing the loan's start and end dates and insert a pipe into the data before the loan identifier, use a marked group as part of the replacement text.

Marked Groups
Use marked groups to leverage data that was identified by a regular expression.
In the expression below, insert a marked group with an open and closed parentheses around the regular expression within the quotes. Then, click Submit.

The replacement text should include the data captured by the marked group. In the replacement text parameter, insert the syntax for a marked group, $1, in front of the pipe and within the quotes. Then, click Submit.

After running the workflow, the output now includes the start and end dates of the loans, which was captured by the marked group, and a pipe delimiter to indicate the start of the loan identifier.

Matching with RegEx
In the Formula tool, create another column called "Match" to evaluate whether a row of data matches a specific pattern: the presence of a correctly formatted loan start date, comma, and correctly formatted loan end date, followed by the additional information. Begin to create the expression by opening the function library.

Then, expand the string functions and select "RegEx_Match" from the list.

The RegEx_Match function requires two parameters: the string column in which to identify matches, [Column 1], and a regular expression to define a pattern: the correctly formatted start and end dates of the loan, separated by a comma and followed by the remaining data.

After running the workflow, a new column called [Match] evaluates whether the input matches the pattern specified by our regular expression. Rows that do match the regular expression contain the value negative 1 in the column, while rows whose values do not match the regular expression contain a value of 0.

Count Matches

Now, count the number of times a matching pattern of a correctly formatted date is identified in the data. Each row should include two dates: one for the start of the loan and another for the end of the loan. Create a new column called [Number of Matches] and use the function RegEx_CountMatches to output the number of times the expected date pattern is found in each row of data. In the expression editor, insert the values of the parameters to use for this function: [Column1] as the column of data to analyze, and the expected pattern of a date in this data: a four digit year, hyphen, two digit month, hypen, two digit day, all in quotes.

Complete and valid data contains values that match this pattern twice. Sure enough, the results indicate that nearly every row of data does conform to this format. The value in the column [Number of Matches] is two (2), except for one row where a match was not found, as indicated by the value of zero (0).