Multi-Row Formulas

The Multi-Row Formula Tool

While the grass isn't always greener on the next row, sometimes it's useful to refer to a value from the row above or below the current value to make calculations. Whether you are trying to create a running total, parse complex data, or some perform some other mathematical calculations, the multi-row formula tool makes it possible.

Multi-Row vs Formula

The multi-row formula is very similar to the standard formula tool in that you can create a new column or modify an existing column. Both tools have expression editors, making it easy to use functions to manipulate values. Both tools process values from the top down, meaning that expressions are applied to rows in order. Where the tools differ is in the ability to look across rows and support multiple expressions in a single tool.

While the formula tool is limited to using values contained in the current row being processed, the multi-row formula is not. However, unlike the formula tool, multi-row formulas are limited to one expression per tool. While both tools are very versatile, the multi-row formula offers flexibility that the formula tool cannot.

A dataset contains information on monthly reports, which year the report was generated, and total sales for that month. Not every month has a value in the column Year. Drag a Multi-Row Formula tool onto the canvas to fill in the missing values.

Fill Til You're Full

Data is typically filled in two ways. The first is by replicating the same value over and over. The second way to fill is using a series, or filling cells in a graduated manor. In this instance, the value associated with Report A should be replicated until the next year's Report A.

In the Multi-row formula's configuration window, select "Update Existing Field" and use the dropdown to select the column "Year".

Select the Functions tab and expand the "Conditional" category.

Clicking once shows a description of the formula here.

Insert the first conditional statement by double clicking it.

Click the "Variables" tab.

In the Multi-row formula's expression editor, the variables tab includes not only the columns for the current row being evaluated, but the rows above and the rows below. Use these variables to complete the conditional statement.

In place of the "condition" placeholder, use the IsNull function to test if the row contains a value in the column "Year".

In place of the "true" placeholder, insert the previous row's value from the column Year by expanding the category for the previous row, then double click the column Year.

In place of the "false" placeholder, keep the existing value by expanding the category for the active row and double clicking Year.

After running the workflow, the null values in the column [Year] have been filled.

Group By
Now identify the month each report represents using the multi-row formula. A new column called Month is created. For each year, the month should be the previous rows value plus 1. Because the count starts over when the year changes, "Group By: Year" is selected. This will apply the function as long as the value in the column being grouped is the same as the row before. When the value differs, the formula will begin the counting sequence over again. The expression for this tool is the previous row's value in the column Month plus 1. After running the workflow, the values in the column Month count from 1 through 12, then start again when the year changes.

+1, -1, and Beyond
Using another Multi-row formula, compare the total sales to the same month from the previous year. Create a new field entitled "Sales % vs Previous Year".

Use the dropdown to change the column type to float.

Change the number of rows to 12 to see a full year ahead or behind.

Divide the active row's value for total sales by the value in total sales 12 rows before. Multiply that value by 100 to generate percentages. After running the workflow, the first 12 rows contain null values. This is because the expression is looking for a value 12 rows before the active row which does not exist. When no value is found, the expression uses a 0. Division by 0 is not possible and a null value is returned.

Closest Valid Row
Finally, use the multi-row formula tool to compare total sales to the previous month's total sales. Create a new field entitled "Monthly sales diff". In this expression, subtract the

previous row's total sales from the active row's value. After running the workflow, the values appear correct, but the first row returned the value in the current row. As before, when the previous row was not found, a value of 0 was used. To correct this, use the dropdown to change how Designer computes values for rows that don't exist. Select "Set to Values of Closest Valid Row" to have Designer use the uppermost value in the specified column. This will only occur when the specified row is not valid. After running the workflow again, the first row's value is subtracted from itself, returning a value of 0.