EXAMPLE - Comparison Functions2

In the town of Circleville, citizens are allowed to maintain a single crop circle in their backyard, as long as it confirms to the town regulations. Below is some data on the size of crop circles in town, with a separate entry for each home. Limits are displayed in the adjacent columns, with the `inclusive` columns indicating whether the minimum or maximum values are inclusive.

Tip: As part of this exercise, you can see how to you can extend your recipe to perform some simple financial analysis of the data.

Source:

Location	Radius_ft	minRadius_ft	minInclusive	maxRadius_ft	maxInclusive
House1	55.5	10	Y	25	N
House2	12	10	Y	25	N
House3	14.25	10	Y	25	N
House4	3.5	10	Y	25	N
House5	27	10	Y	25	N

Transform:

After the data is loaded into the Transformer page, you can begin comparing column values:

```
derive type:single value: LESSTHANEQUAL(Radius_ft,minRadius_ft) as:'tooSmall'
```

While accurate, the above transform does not account for the `minInclusive` value, which may be changed as part of your steps. Instead, you can delete the previous transform and use the following, which factors in the other column:

```
derive type:single value: IF(minInclusive == 'Y',LESSTHANEQUAL(Radius_ft,minRadius_ft),LESS
THAN(Radius_ft,minRadius_ft)) as:'tooSmall'
```

In this case, the `IF` function tests whether the minimum value is inclusive (values of `10` are allowed). If so, the `LE SSTHANEQUAL` function is applied. Otherwise, the `LESSTHAN` function is applied. For the maximum limit, the following step applies:

```
derive type:single value: IF(maxInclusive == 'Y',GREATERTHANEQUAL(Radius_ft,maxRadius_ft),
GREATERTHAN(Radius_ft,maxRadius_ft)) as:'tooBig'
```

Now, you can do some analysis of this data. First, you can insert a column containing the amount of the fine per foot above the maximum or below the minimum. Before the first `derive` command, insert the following, which is the fine (`$15.00`) for each foot above or below the limits:

```
derive type:single value: 15 as:'fineDollarsPerFt'
```

At the end of the recipe, add the following new line, which calculates the fine for crop circles that are too small:

```
derive type:single value: IF(tooSmall == 'true', (minRadius_ft - Radius_ft) *
fineDollarsPerFt, 0.0) as: 'fine_Dollars'
```

The above captures the too-small violations. To also capture the too-big violations, change the above to the following:

```
derive type:single value: IF(tooSmall == 'true', (minRadius_ft - Radius_ft) *
fineDollarsPerFt, if(tooBig == 'true', (Radius_ft - maxRadius_ft) * fineDollarsPerFt,
'0.0')) as: 'fine_Dollars'
```

In place of the original "false" expression (`0.0`), the above adds the test for the too-big values, so that all fines are included in a single column. You can reformat the `fine_Dollars` column to be in dollar format:

```
set col: fine_Dollars value: NUMFORMAT(fine_Dollars, '$###.00')
```

Results:

After you drop the columns used in the calculation and move the remaining ones, you should end up with a dataset similar to the following:

Location	fineDollarsPerFt	Radius_ft	minRadius_ft	minInclusive	maxRadius_ft	maxInclusive	fineDollars
House1	15	55.5	10	Y	25	N	$457.50
House2	15	12	10	Y	25	N	$0.00
House3	15	14.25	10	Y	25	N	$0.00
House4	15	3.5	10	Y	25	N	$97.50
House5	15	27	10	Y	25	N	$30.00

Now that you have created all of the computations for generating these values, you can change values for `minRadius_ft`, `maxRadius_ft`, and `fineDollarsPerFt` to analyze the resulting fine revenue. Before or after the transform where you set the value for `fineDollarsPerFt`, you can insert something like the following:

```
set col: minRadius_ft value:'12.5'
```

After the step is added, select the last line in the recipe. Then, you can see how the values in the `fineDollars` column have been updated.