ROLLINGMODE Function

Contents:

Computes the rolling mode (most common value) forward or backward of the current row within the specified column.

- If an input value is missing or null, it is not factored in the computation. For example, for the first row in the dataset, the rolling mode of previous values is undefined.
- The row from which to extract a value is determined by the order in which the rows are organized based on the `order` parameter.

  If you are working on a randomly generated sample of your dataset, the values that you see for this function might not correspond to the values that are generated on the full dataset during job execution.
- The function takes a column name and two optional integer parameters that determine the window backward and forward of the current row.
  - The default integer parameter values are `-1` and `0`, which computes the rolling function from the current row back to the first row of the dataset.
- This function works with the following transforms:
  - *Window Transform*
  - *Set Transform*
  - *Derive Transform*

For more information on a non-rolling version of this function, see *MODE Function*.

Basic Usage

Column example:

```
derive type:single value:ROLLINGMODE(myCol)
```

Output: Generates a new column containing the rolling mode of all values in the `myCol` column from the first row of the dataset to the current one.

Rows before example:

```
window value:ROLLINGMODE(myNumber, 3)
```

Output: Generates the new column, which contains the rolling mode of the current row and the three previous row values in the `myNumber` column.

Rows before and after example:

```
window value:ROLLINGMAX(myNumber, 3, 2)
```

Output: Generates the new column, which contains the rolling mode of the three previous row values, the current row value, and the two rows after the current one in the `myNumber` column.

Syntax and Arguments

```
window value:ROLLINGMODE(col_ref, rowsBefore_integer, rowsAfter_integer) order: order_col
[group: group_col]
```

Argument	Required?	Data Type	Description
col_ref	Y	string	Name of column whose values are applied to the function
rowsBefore_integer	N	integer	Number of rows before the current one to include in the computation
rowsAfter_integer	N	integer	Number of rows after the current one to include in the computation

For more information on the `order` and `group` parameters, see *Window Transform*.

For more information on syntax standards, see *Language Documentation Syntax Notes*.

col_ref

Name of the column whose values are used to compute the function.

- Multiple columns and wildcards are not supported.

Usage Notes:

Required?	Data Type	Example Value
Yes	String (column reference to Integer or Decimal values)	`myColumn`

rowsBefore_integer, rowsAfter_integer

Integers representing the number of rows before or after the current one from which to compute the rolling function, including the current row. For example, if the first value is `5`, the current row and the five rows before it are used in the computation. Negative values for `k` compute the rolling average from rows preceding the current one.

- `rowBefore=0` generates the current row value only.
- `rowBefore=-1` uses all rows preceding the current one.
- If `rowsAfter` is not specified, then the value `0` is applied.
- If a `group` parameter is applied, then these parameter values should be no more than the maximum number of rows in the groups.

Usage Notes:

Required?	Data Type	Example Value
No	Integer	4

Examples

Tip: For additional examples, see *Common Tasks*.

Example - Counting most common coin flips

In the following table, 20 coin flips are tabulated. You want to capture a rolling evaluation of the most common value.

Source:

Turn	Result
1	heads
2	heads
3	tails
4	heads
5	heads
6	tails
7	tails
8	heads
9	tails
10	heads
11	tails
12	heads
13	tails
14	heads
15	heads
16	tails
17	tails
18	tails
19	heads
20	heads

Transform:

To use the `ROLLINGMODE` function, the results need to be converted to numeric values:

```
set col: Result value: if(Result == 'heads', 0, 1)
```

Now calculate the `ROLLINGMODE` for the preceding five values for each row:

```
derive type: multiple value: rollingmode(Result, 4, 0) order: Turn as: 'mostCommonLast5'
```

You can now convert the binary values back to text information:

```
set col: mostCommonLast5 value: if(mostCommonLast5 == 0, 'heads-last5', 'tails-last5')
```

Results:

Turn	Result	mostCommonLast5
1	heads	heads-last5

2	heads	heads-last5
3	tails	heads-last5
4	heads	heads-last5
5	heads	heads-last5
6	tails	heads-last5
7	tails	tails-last5
8	heads	heads-last5
9	tails	tails-last5
10	heads	tails-last5
11	tails	tails-last5
12	heads	heads-last5
13	tails	tails-last5
14	heads	heads-last5
15	heads	heads-last5
16	tails	heads-last5
17	tails	tails-last5
18	tails	tails-last5
19	heads	tails-last5
20	heads	tails-last5