

# SIGN Function

Computes the positive or negative sign of a given numeric value. The value can be a Decimal or Integer literal, a function returning Decimal or Integer, or a reference to a column containing numeric values.

- For positive values, this function returns 1.
- For negative values, this function returns -1.
- For the value 0, this function returns 0.

## Basic Usage

### Column reference example:

```
derive type:single value:SIGN(MyInteger) as:'sign_MyInteger'
```

**Output:** Generates the new `sign_MyInteger` column containing the absolute value of each value found in the `MyInteger` column.

### Numeric literal example:

```
delete row:(SIGN(MyInteger) == -1)
```

**Output:** Deletes all rows from the dataset in which the sign of the entry in the `MyInteger` column is -1. This transform remove all rows contain negative values in the `MyInteger` column.

## Syntax and Arguments

```
derive type:single value:SIGN(numeric_value)
```

Argument	Required?	Data Type	Description
numeric_value	Y	Decimal or Integer	Decimal or Integer literal, function returning Decimal or Integer, or name of column to apply to the function

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

### numeric\_value

Numeric literal, function returning numeric literal, or name of the column containing values the sign of which are to be computed.

- Missing input values generate missing results.
- Literal numeric values should not be quoted. Quoted values are treated as strings.
- Multiple columns and wildcards are not supported.

### Usage Notes:

Required?	Data Type	Example Value
Yes	String (column reference) or Integer or Decimal value	-10.5

## Examples

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

### Example - Basic SIGN function

#### Source:

Your source data looks like the following, which measures coordinate distances from a fixed point on a grid:

X	Y
-2	4
-6.2	-2
0	-4.2
4	4
15	-0.05

#### Transform:

You can use the following transform to derive the sign values of these columns:

```
derive type:single value: SIGN(X) as: 'signX'
```

```
derive type:single value: SIGN(Y) as: 'signY'
```

Using these two columns, you can assign each set of coordinates into a quadrant:

```
case cases: [(signX == 1) && (signY == -1), 'lower-right'], [(signX == 1) && (signY == 1), 'upper-right'], [(signX == -1) && (signY == -1), 'lower-left'], [(signX == -1) && (signY == 1), 'upper-left'] default: 'line' as: 'quadrant'
```

#### Results:

X	Y	signX	signY	quadrant
-2	4	-1	1	upper-left
-6.2	-2	-1	-1	lower-left
0	-4.2	0	-1	line
4	4	1	1	upper-right
15	-0.05	1	-1	lower-right