

ACOS Function

For input values between -1 and 1 inclusive, this function returns the angle in radians whose cosine value is the input. This function is the inverse of the cosine function. The value can be a Decimal or Integer literal or a reference to a column containing numeric values.

NOTE: While this function returns values outside of the range $-1 \leq x \leq 1$, those values are not considered valid.

For more information on the cosine function, see *COS Function*.

Arc secant:

The arc secant function is computed as follows:

Input Range	Output computation
$x \leq -1$	$\text{acos}(1/x)$
$x \geq 1$	$\text{acos}(1/x)$
$-1 < x < 1$	Not valid

Wrangle vs. SQL: This function is part of Wrangle, a proprietary data transformation language. Wrangle is not SQL. For more information, see *Wrangle Language*.

Basic Usage

Numeric literal example:

```
acos(0.5)
```

Output: Returns the computation of the arc cosine of 0.5. Output value is in radians.

Column reference example:

```
acos(X)
```

Output: Returns the arc cosine of the values in `x` column.

Syntax and Arguments

```
acos(numeric_value)
```

Argument	Required?	Data Type	Description
numeric_value	Y	string, decimal, or integer	Name of column, Decimal or Integer literal, or function returning those types to apply to the function

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

numeric_value

Name of the column, Integer or Decimal literal, or function returning that data type to apply to the function.

- 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 literal	0 . 5

Examples

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

Example - Trigonometry Arc functions

This example illustrates how to apply the inverse trigonometric (Arc) functions to your transformations.

NOTE: These functions are valid over specific ranges.

- **Arcsine.** See *ASIN Function*.
- **Arccosine.** See *ACOS Function*
- **Arctangent.** See *ATAN Function*.
- **Arccotangent.** Computed using ATAN function. See below.
- **Arcsecant.** Computed using ACOS function. See below.
- **Arccosecant.** Computed using ASIN function. See below.

Source:

In the following sample, input values are in radians. In this example, all values are rounded to two decimals for clarity.

Y
-1.00
-0.75
-0.50
0.00
0.50
0.75
1.00

Transformation:

Arcsine:

Valid over the range (-1 <= Y <= 1)

Transformation Name	New formula

Parameter: Formula type	Single row formula
Parameter: Formula	<code>round(degrees(asin(Y)), 2)</code>
Parameter: New column name	'asinY'

Arccosine:

Valid over the range $(-1 \leq Y \leq 1)$

Transformation Name	New formula
Parameter: Formula type	Single row formula
Parameter: Formula	<code>round(degrees(acos(Y)), 2)</code>
Parameter: New column name	'acosY'

Arctangent:

Valid over the range $(-1 \leq Y \leq 1)$

Transformation Name	New formula
Parameter: Formula type	Single row formula
Parameter: Formula	<code>round(degrees(atan(Y)), 2)</code>
Parameter: New column name	'atanY'

Arccosecant:

This function is valid over a set of ranged inputs, so you can use a conditional column for the computation. For more information, see *ASIN Function*.

Transformation Name	Conditional column
Parameter: Condition type	if...then...else
Parameter: If	<code>(Y <= -1) (Y >= 1)</code>
Parameter: Then	<code>round(degrees(asin(divide(1, Y))), 2)</code>
Parameter: New column name	'acscY'

Arcsecant:

Same set of ranged inputs apply to this function. For more information, see *ACOS Function*.

Transformation Name	Conditional column
Parameter: Condition type	if...then...else
Parameter: If	<code>(Y <= -1) (Y >= 1)</code>
Parameter: Then	<code>round(degrees(acos(divide(1, Y))), 2)</code>

Parameter: New column name	'asecY'
-----------------------------------	---------

Arccotangent:

For this function, the two different ranges of inputs have different computations, so an `else` condition is added to the transformation. For more information, see *ATAN Function*.

Transformation Name	Conditional column
Parameter: Condition type	if...then...else
Parameter: If	$Y > 0$
Parameter: Then	<code>round(degrees(atan(divide(1, Y))), 2)</code>
Parameter: Else	<code>round(degrees(atan(divide(1, Y)) + pi()), 2)</code>
Parameter: New column name	'acotY'

Results:

Y	acotY	asecY	acscY	atanY	acosY	asinY
-1.00	-41.86	180.00	-90.00	-45.00	180.00	-90.00
-0.75	-49.99	null	null	-37.00	139.00	-49.00
-0.50	-60.29	null	null	-27.00	120.00	-30.00
0.00	null	null	null	0.00	90.00	0.00
0.50	63.44	null	null	27.00	60.00	30.00
0.75	53.13	null	null	37.00	41.00	49.00
1.00	45.00	0.00	90.00	45.00	0.00	90.00