

# PI Function

The `PI` function generates the value of pi to 15 decimal places: 3.1415926535897932.

This function uses no parameters. Generated values are of Decimal type and have fifteen digits of precision after the decimal point. If you want to see a fewer number of digits in the generated value, you might need to apply a different number format or using the `ROUND` function.

- See *NUMFORMAT Function*.
- See *ROUND Function*.

**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

```
pi()
```

**Output:** Returns the value of pi.

## Syntax and Arguments

There are no arguments for this function.

## Examples

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

## Example - pi column

In the following example, the source is simply the `source` column, and the `pi` column is generated by the `PI` function:

<b>Transformation Name</b>	New formula
<b>Parameter: Formula type</b>	Single row formula
<b>Parameter: Formula</b>	pi()
<b>Parameter: New column name</b>	'pi'

source	pi
A	3.1415926535897932
B	3.1415926535897932
C	3.1415926535897932
D	3.1415926535897932

## Example - RANDBETWEEN, PI, and TRUNC functions

This example illustrates how you can apply the following functions to generate new and random data in your dataset:

- **RANDBETWEEN** - Generate a random Integer value between two specified Integers. See *RANDBETWEEN Function*.
- **PI** - Generate the value of pi to 15 decimal points. See *PI Function*.
- **ROUND** - Round a decimal value to the nearest Integer or to a specified number of digits. See *ROUND Function*.
- **TRUNC** - Round a value down to the nearest Integer value. See *TRUNC Function*.

### Source:

In the following example, a company produces 10 circular parts, the size of which is measured in each product's radius in inches.

prodId	radius_in
p001	1
p002	2
p003	3
p004	4
p005	5
p006	6
p007	7
p008	8
p009	9
p010	10

Based on the above data, the company wants to generate some additional sizing information for these circular parts, including the generation of two points along each part's circumference where quality stress tests can be applied.

### Transformation:

To begin, you can use the following steps to generate the area and circumference for each product, rounded to three decimal points:

<b>Transformation Name</b>	New formula
<b>Parameter: Formula type</b>	Single row formula
<b>Parameter: Formula</b>	<code>ROUND(PI() * (POW(radius_in, 2)), 3)</code>
<b>Parameter: New column name</b>	'area_sqin'

<b>Transformation Name</b>	New formula
<b>Parameter: Formula type</b>	Single row formula
<b>Parameter: Formula</b>	<code>ROUND(PI() * (2 * radius_in), 3)</code>

<b>Parameter: New column name</b>	'circumference_in'
-----------------------------------	--------------------

For quality purposes, the company needs two tests points along the circumference, which are generated by calculating two separate random locations along the circumference. Since the `RANDBETWEEN` function only calculates using Integer values, you must first truncate the values from `circumference_in`:

<b>Transformation Name</b>	New formula
<b>Parameter: Formula type</b>	Single row formula
<b>Parameter: Formula</b>	<code>TRUNC(circumference_in)</code>
<b>Parameter: New column name</b>	'trunc_circumference_in'

Then, you can calculate the random points using the following:

<b>Transformation Name</b>	New formula
<b>Parameter: Formula type</b>	Single row formula
<b>Parameter: Formula</b>	<code>RANDBETWEEN(0, trunc_circumference_in)</code>
<b>Parameter: New column name</b>	'testPt01_in'

<b>Transformation Name</b>	New formula
<b>Parameter: Formula type</b>	Single row formula
<b>Parameter: Formula</b>	<code>RANDBETWEEN(0, trunc_circumference_in)</code>
<b>Parameter: New column name</b>	'testPt02_in'

## Results:

After the `trunc_circumference_in` column is dropped, the data should look similar to the following:

prodlid	radius_in	area_sq_in	circumference_in	testPt01_in	testPt02_in
p001	1	3.142	6.283	5	5
p002	2	12.566	12.566	3	3
p003	3	28.274	18.850	13	13
p004	4	50.265	25.133	24	24
p005	5	78.540	31.416	0	0
p006	6	113.097	37.699	15	15
p007	7	153.938	43.982	11	11
p008	8	201.062	50.265	1	1
p009	9	254.469	56.549	29	29
p010	10	314.159	62.832	21	21