

How to use the “interpolate” transform function

With an example to interpolate data in a graph.

The “interpolate” transform function creates interpolated y values for an xy data table.

For each “new x value” in range, the corresponding new y value is interpolated from the position of this new x value in the original x column.

New x = 1,5 is in the middle between old x= 1 and old x = 2.

The middle of the corresponding old y values (1 and 4) is 2,5.

New y (x = 1,5) = 2,5

	1-x_range	2-x	3-y	4-interpolated_y	
1	0,0000	0,0000	0,0000	0,0000	
2	0,5000	1,0000	1,0000	0,5000	
3	1,0000	2,0000	4,0000	1,0000	
4	1,5000			2,5000	
5	2,0000			4,0000	
6					

interpolate

The interpolate function performs linear interpolation on a set of X,Y pairs defined by an x range and a y range. The function returns a range of interpolated y values from a range of values between the minimum and maximum of the x range.

Syntax

`interpolate(x range; y range; range)`

Values in the x range argument must be strictly increasing or strictly decreasing.

The range argument must be a single range (indicated with the {} brackets) or a worksheet column. Missing values and text strings are not allowed in the x range and y range. Text strings in range are replaced by missing values.

Extrapolation is not possible; missing value symbols are returned for range argument values less than the lowest x range value or greater than the highest x range value.

Example

For x = {0;1;2}, y = {0;1;4}, and range = data(0; 2; 0,5) (this data operation returns numbers from 0 to 2 at increments of 0,5), the operation `col(1) = interpolate(x; y; range)` places the range {0,0; 0,5; 1,0; 2,5; 4,0} into column 1.

If range had included values outside the range for x, missing values would have been returned for those out-of-range values.

```

x = {0; 1; 2}
y = {0; 1; 4}
x_range = data(0; 2; 0,5)
interpolated_y = interpolate(x; y; x_range)

col(1) = x_range
col(2) = x
col(3) = y
col(4) = interpolated_y
cell(1;0) = "x_range"
cell(2;0) = "x"
cell(3;0) = "y"
cell(4;0) = "interpolated_y"

```

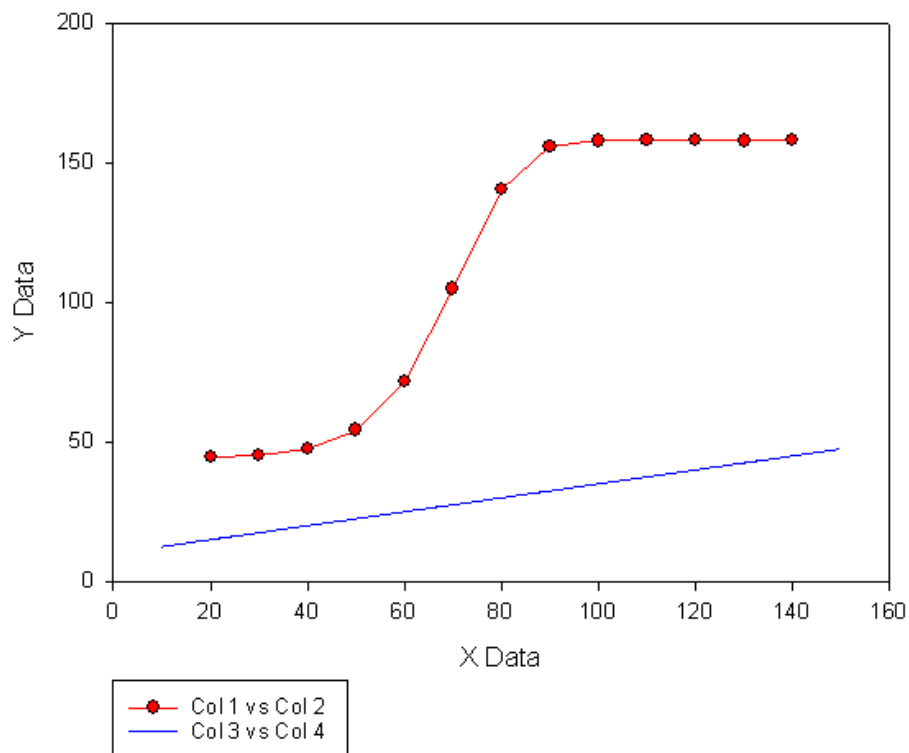
[\[Get Code\]](#)

	1-x_range	2-x	3-y	4-interpolated_y
1	0,0000	0,0000	0,0000	0,0000
2	0,5000	1,0000	1,0000	0,5000
3	1,0000	2,0000	4,0000	1,0000
4	1,5000			2,5000
5	2,0000			4,0000
6				

Interpolating Graph Data

This graph shows two XY pair plots, a (red) line/scatter plot with X intervals of 10, and a (blue) line plot with X intervals of 0,3. To calculate the Y difference of both curves at the X values of the red plot, you need the Y values of the blue curve at the “red” X values. Use the interpolate function to calculate them.

Two XY pair plots in one graph



X red: col(1)

Y red: col(2)

X blue: col(3)

Y blue: col(4)

Run this transform (On the worksheet, press F10, paste the lines into the edit window with Ctrl-V, and click on Run) :

```
Xred = col(1)
```

```
Xblue = col(3)
```

```
Yblue = col(4)
```

```
interpolated_y = interpolate(Xblue; Yblue; Xred)
```

```
col(5) = interpolated_y
```

Note:

Use of list separator (; or ,) and decimal sign (, or .) in a transform.

This depends on your Windows Control Panel settings for:

Region and Language > Formats > Additional settings > Numbers.

Valid combinations are:

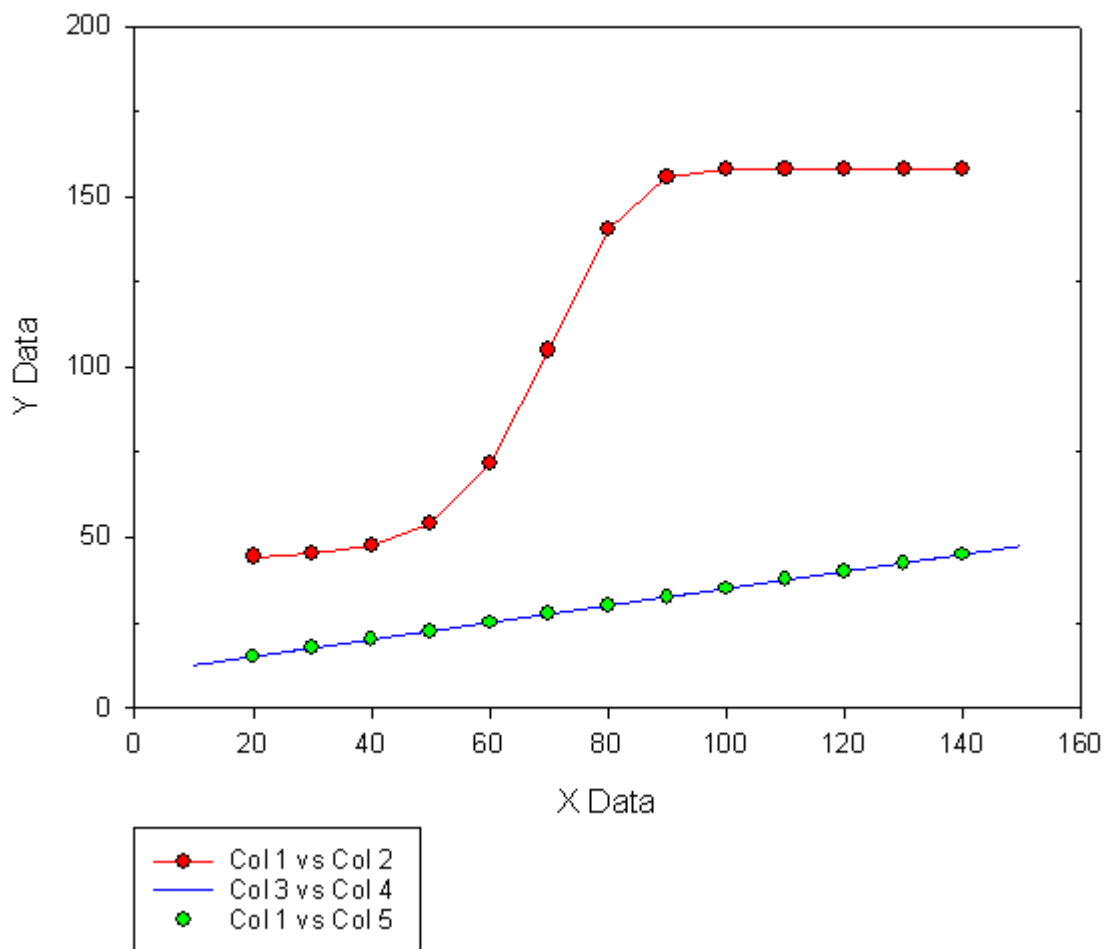
- Semicolon and comma: list separator (;) and decimal sign (,)
- Comma and point: list separator (,) and decimal sign (.)

Details:

<http://www.systat.de/TT201205/DecimalsignSigmaPlot.pdf>

Now add a scatter plot with the new Y data: Graph Page > Add Plot.

Three XY pair plots in one graph



P.S.: Sample data

To use the sample data from this example, run these lines in a transform:

```
col(1) = data(20; 140; 10)
col(2) = {44,4; 45,2; 47,6; 54,3; 71,5; 104,9; 140,7; 155,7; 157,9; 158,0;
158,0; 157,9; 158,1}

col(3) = data(10;150; 0,3)
x=col(3)
y=0,25 * x + 10
col(4) = y
```