The dcolumn package[*]

David Carlisle

2001/05/28

Abstract

This package defines a system for defining columns of entries in an `array` or `tabular` which are to be aligned on a ‘decimal point’.

This package defines `D` to be a column specifier with three arguments.

`D{`⟨*sep.tex*⟩`}{`⟨*sep.dvi*⟩`}{`⟨*decimal places*⟩`}`

⟨*sep.tex*⟩ should be a single character, this is used as the separator in the `.tex` file. Thus it will usually be ‘`.`’ or ‘`,`’.

⟨*sep.dvi*⟩ is used as the separator in the output, this may be the same as the first argument, but may be any math-mode expression, such as `\cdot`. It should be noted that `dcolumn` always uses math mode for the digits as well as the separator.

⟨*decimal places*⟩ should be the maximum number of decimal places in the column. If this is negative, any number of decimal places can be used in the column, and all entries will be centred on (the leading edge of) the separator. Note that this can cause a column to be too wide, compare the first two columns in the example below. If this argument is positive, the column uses macros equivalent to `\rightdots` `\endrightdots` of `array.sty`, otherwise the macros are essentially equivalent to `\centerdots` `\endcenterdots`.

You may not want to use all three entries in the `array` or `tabular` preamble, so you may define your own preamble specifiers using `\newcolumntype`.

For example we may say:

`\newcolumntype{d}[1]{D{.}{\cdot}{#1}}`

`d` takes a single argument specifying the number of decimal places, and the `.tex` file should use `.`, with · being used in the output.

`\newcolumntype{.}{D{.}{.}{-1}}`

`.` specifies a column of entries to be centred on the ..

`\newcolumntype{,}{D{,}{,}{2}}`

`,` specifies takes a column of entries with at most two decimal places after a ,.

The following table begins `\begin{tabular}{|d{-1}|d{2}|.|,|}`

1·2	1·2	1.2	1,2
1·23	1·23	12.5	300,2
1121·2	1121·2	861.20	674,29
184	184	10	69
·4	·4		,4
		.4	

[*] This file has version number v1.06, last revised 2001/05/28.

Note that the first column, which had a negative ⟨*decimal places*⟩ argument is wider than the second column, so that the decimal point appears in the middle of the column. Also note that this package deals correctly with entries with no decimal part, no integer part, and blank entries.

If you have table headings (inserted with `\multicolumn{1}{c}{..}` to over-ride the `D` column type) then it may be that neither of the above 'centred' or 'right aligned' forms is quite what you want.

head	head	head
1.2	1.2	1.2
11212.2	11212.2	11212.2
.4	.4	.4

wide heading	wide heading	wide heading
1.2	1.2	1.2
.4	.4	.4

In both of these tables the first column is set with `D{.}{.}{-1}` to produce a column centered on the `.`, and the second column is set with `D{.}{.}{1}` to produce a right aligned column.

The centered column produces columns that are wider than necessary to fit in the numbers under a heading as it has to ensure that the decimal point is centred. The right aligned column two does not have this drawback, but under a wide heading a column of small right aligned figures looks a bit odd.

In version v1.03 a third possibility is introduced. The third ⟨*decimal places*⟩ argument may specify *both* the number of digits to the left and to the right of the decimal place. The third column in the first table above is set with `D{.}{.}{5.1}` and in the second table, `D{.}{.}{1.1}`, to specify 'five places to the left and one to the right' and 'one place to the left and one to the right' respectively. (You may use ',' or other tokens, not necessarily '.' in this argument.) The column of figures is then positioned such that a number with the specified numbers of digits is centred in the column.

This notation also enables columns that are centred on the mid-point of the separator, rather than its leading edge; for example `D{+}{\,\pm\,}{3,3}` will give nice, symmetric layout of up to three digits on either side of a ± sign.