

The `warpcol` package*

W. A. Rochester

November 21, 2007

Abstract

This file defines a tabular column type for formatting numerical columns. The column type enables numerical items to be right justified relative to each other, while centred beneath the column label. In addition, macros are provided to enable variations on this column type to be defined.

1 Introduction

The primary aim of this package is to provide a means to centre a tabular column label over a column of right-justified numbers. However, the package goes further to provide a general mechanism for justifying numerical items relative to each other and column labels. The package works by justifying each numerical item within a box whose width is specified by a ‘pattern’. The width is calculated as the width of the pattern after each number in the pattern is expanded to that number of zeros. For example, the pattern ‘-3.2’ would be expanded to ‘-000.00’. Naturally, pattern width calculation and typesetting of numerical items are performed in math mode.

The applications of this package overlap with those of David Carlisle’s `dcolumn` package, which aligns numbers on their decimal points. However, the differing approaches of the packages create a number of differences in the ways they can be used. First, the `dcolumn` package can align items with varying numbers of decimal places. To achieve such alignment with this package, numbers must be padded with, for example, the active character `~`, redefined to print a space the width of a digit. Second, the present package can align items that are not simple numbers. Such items include negative numbers, which are not properly handled by the `dcolumn` package (but can be if the package is modified slightly).

2 Usage

The package defines the `P` column type for use with the most common application of the package: centring labels over right-justified numbers. This column type is

*This file has version number v1.0c, last revised 2007/11/21 for adding LPPL license statement.

used as follows:

```
\begin{tabular}{P{3.1}P{-2.1}} \hline\hline
\multicolumn{1}{c}{Label 1} & \multicolumn{1}{c}{Label 2} \\ \hline
123.4 & -12.3 \\
12.3 & 12.3 \\
1.2 & 1.2 \\ \hline\hline
\end{tabular}
```

Label 1	Label 2
123.4	-12.3
12.3	12.3
1.2	1.2

The column labels must be set using the `\multicolumn` command to prevent them from being formatted with the P column type. Additional column types can be defined by the use of the `\pcolbegin` and `\pcolend` commands. The `\pcolbegin` command takes two arguments. The first is a code (l, c or r) specifying how numerical items should be justified relative to each other. The second is the pattern. The remaining examples make use of the following definitions (which are not included in the package):

```
\newcolumnntype{L}[1]{>\pcolbegin{r}{#1}}l<\pcolend}
\newcolumnntype{R}[1]{>\pcolbegin{r}{#1}}r<\pcolend}
\newcommand{\pcolhd}[1]{\multicolumn{1}{c}{#1}}
```

Column labels are justified correctly regardless of whether they are shorter or longer than the numerical items:

```
\begin{tabular}{L{3.1}L{3.1}P{3.1}P{3.1}R{3.1}R{3.1}} \hline\hline
\multicolumn{1}{l}{SL} & \multicolumn{1}{l}{Long label} & & & & & \\
\multicolumn{1}{c}{SL} & \multicolumn{1}{c}{Long label} & & & & & \\
\multicolumn{1}{r}{SL} & \multicolumn{1}{r}{Long label} \\ \hline\hline
123.4 & 123.4 & 123.4 & 123.4 & 123.4 & 123.4 \\
12.3 & 12.3 & 12.3 & 12.3 & 12.3 & 12.3 \\
1.2 & 1.2 & 1.2 & 1.2 & 1.2 & 1.2 \\ \hline\hline
\end{tabular}
```

SL	Long label	SL	Long label	SL	Long label
123.4	123.4	123.4	123.4	123.4	123.4
12.3	12.3	12.3	12.3	12.3	12.3
1.2	1.2	1.2	1.2	1.2	1.2

Numerical items can be left justified, right justified or centred relative to column labels:

```

\begin{tabular}{L{3.1}R{3.1}P{3.1}} \hline\hline
\pcolhd{Left justified} & \pcolhd{Right justified} & \pcolhd{Centred}
\\ \hline
123.4 & 123.4 & 123.4 \\
12.3 & 12.3 & 12.3 \\
1.2 & 1.2 & 1.2 \\
\end{tabular}

```

Left justified	Right justified	Centred
123.4	123.4	123.4
12.3	12.3	12.3
1.2	1.2	1.2

They can also be left justified, right justified or centred relative to each other:

```

\begin{tabular}{>{\pcolbegin{1}{3.1}}c<{\pcolend}P{3.1}%
>{\pcolbegin{c}{3.1}}c<{\pcolend}} \hline\hline
\pcolhd{Left justified} & \pcolhd{Right justified} & \pcolhd{Centred}
\\ \hline
123.4 & 123.4 & 123.4 \\
12.3 & 12.3 & 12.3 \\
1.2 & 1.2 & 1.2 \\
\end{tabular}

```

Left justified	Right justified	Centred
123.4	123.4	123.4
12.3	12.3	12.3
1.2	1.2	1.2

Finally, here is an example with columns that contain something other than simple numbers:

```

\begin{tabular}{>{\pcolbegin{1}{< 1.3}}c<{\pcolend}P{3\,3.2}}
\hline\hline
\pcolhd{$p$} & \pcolhd{Cost (\$)} \\
> 0.1 & 123\,456.78 \\
< 0.1 & 23\,456.78 \\
< 0.01 & 4\,523.45 \\
< 0.001 & 345.67 \\
\end{tabular}

```

p	Cost (\$)
> 0.1	123 456.78
< 0.1	23 456.78
< 0.01	4 523.45
< 0.001	345.67

3 Implementation

```
1 <*package>
   Load the array package, if it is not already loaded.
2 \RequirePackage{array}
   Declare counters and dimensions.
3 \newcount\WPC@digcnt
4 \newbox\WPC@patbox
5 \newbox\WPC@valbox
\WPC@digval The \WPC@digval command prints the given character if the character is a digit,
and 'x' otherwise.
6 \def\WPC@digval#1{%
7 \ifx#100%
8 \else\ifx#111%
9 \else\ifx#122%
10 \else\ifx#133%
11 \else\ifx#144%
12 \else\ifx#155%
13 \else\ifx#166%
14 \else\ifx#177%
15 \else\ifx#188%
16 \else\ifx#199%
17 \else x\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi}

\WPC@digstr The \WPC@digstr command prints a row of zeroes the length of the count in the
\WPC@digcnt counter.
18 \def\WPC@digstr{%
19 \loop 0\advance\WPC@digcnt by-1 \ifnum\WPC@digcnt>0 \repeat}

\WPC@expandpat The \WPC@expandpat command expands the given pattern, for example from -3.2
to -000.00.
20 \def\WPC@expandpat#1{%
21 \ifx#1\end
22 \ifnum\WPC@digcnt>0 \WPC@digstr\fi
23 \let\WPC@next=\relax
24 \else
25 \edef\WPC@dv{\WPC@digval#1}%
26 \if x\WPC@dv
27 \ifnum\WPC@digcnt>0 \WPC@digstr\fi
28 #1%
29 \else
30 \multiply\WPC@digcnt by10 \advance\WPC@digcnt by\WPC@dv
31 \fi
32 \let\WPC@next=\WPC@expandpat
33 \fi
34 \WPC@next}
```

`\pcolbegin` The `\pcolbegin` command is executed before column entries. The first argument is the justification code (l, r or c), and the second is the pattern (e.g. -2.1). The command begins the definition of a box containing the table item. The width of the box is set to the width of the expanded pattern. Fill commands justify the value within the box. The value is typeset in math mode.

```
35 \def\pcolbegin#1#2{%
```

Initialise the digit count and save the justification code.

```
36 \WPC@digcnt=0
```

```
37 \edef\WPC@just{#1}%
```

Create a box containing the expanded pattern (e.g. -00.0).

```
38 \setbox\WPC@patbox=\hbox{\$ \WPC@expandpat#2 \end\$}%
```

Begin the definition of the value box.

```
39 \setbox\WPC@valbox=\hbox to\wd\WPC@patbox\bgroup
```

```
40 \if#1r\hfil\else\if#1c\hfil\fi\fi$}
```

`\pcolend` The `\pcolend` command is executed after column entries. It completes the box definition started by the `\pcolbegin` command, and then prints the box.

```
41 \newcommand{\pcolend}{\$%
```

```
42 \if 1\WPC@just\hfil\else\if c\WPC@just\hfil\fi\fi
```

```
43 \egroup\box\WPC@valbox}
```

P Declare the P column type.

```
44 \newcolumntype{P}[1]{>\pcolbegin{r}{#1}}c<\pcolend}}
```

```
45 </package>
```