The **tabulary** package*

David Carlisle

2014/06/11

1 User Documentation

\begin{tabulary}{⟨\text{length}\rangle}{⟨\text{pream}\rangle} ... \end{tabulary}

The rather daft name may change in a later release but it is a pun on **tabularx**, which itself was a pun on **tabular**.

These environments work pretty much like the standard tabular environment (or more correctly, the enhanced version from the array package) except that there are more possibilities for the column types.

**LCRJ** These new 'uppercase' column types are only activated in the **tabulary** environment. In order to make the total table width equal to ⟨\text{length}\rangle the LCRJ columns are converted to \texttt{p} columns (with \texttt{\raggedright}, \texttt{\centering}, or \texttt{\raggedleft} or normal justification respectively applied). The width of these converted columns is proportional to the natural width of the longest entry in each column.

To stop very narrow columns being too 'squeezed' by this process any columns that are narrower than \texttt{\tymin} are set to their natural width. This length may be set with \texttt{\setlength} and is arbitrarily initialised to 10 pt. (If you know that a column will be narrow, it may be preferable to use, say, \texttt{c} rather than \texttt{C} so that the **tabulary** mechanism is never invoked on that column.)

Similarly one very large entry can force its column to be too wide. So to prevent this, all columns with natural length greater than \texttt{\tymax} are set to the same width (with the proportion being taken as if the natural length was equal to \texttt{\tymax}). This is initially set to twice the text width.

Narrow \texttt{p} columns are sometimes quite hard to set, and so you may redefine the command \texttt{\tyformat} to be any declarations to make just after the \texttt{\centering} or \texttt{\ragged}... declaration. By default it redefines \texttt{\everypar} to insert a zero space at the start of every paragraph, so the first word may be hyphenated. (See **DogBook**).

As the environment makes a standard \LaTeX box, it will be indented by the paragraph indent at the start of a paragraph, and so will not fit on a line if

*This file has version number v0.10, last revised 2014/06/11.
given argument \textwidth unless it is preceded by \noindent or is in a center environment or some other environment with zero paragraph indent.

2 Features

You can use \multicolumn but if the multicolumn text turns out to be longer than the final calculated widths of the columns that it spans, then the final table will be too wide.

\verb does not work. (except in restricted version as in tabularx)

The whole table is evaluated twice, so take care with some \TeX constructions that may have side effects like writing to files.

3 Options

The following package option is defined:

debugshow Causes a lot of stuff to appear on the terminal. I find this invaluable, you may find it less so.
## Examples

<table>
<thead>
<tr>
<th></th>
<th>With C columns</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>the rain in spain (an @ expr.) the rain in spain falls mainly on the plain the rain in spain falls mainly on the plain</td>
</tr>
<tr>
<td>a</td>
<td>b (an @ expr.) c</td>
</tr>
<tr>
<td>a</td>
<td>b b (an @ expr.) c c</td>
</tr>
<tr>
<td>a</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>With J columns</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>the rain in spain (an @ expr.) the rain in spain falls mainly on the plain the rain in spain falls mainly on the plain</td>
</tr>
<tr>
<td>a</td>
<td>b (an @ expr.) c</td>
</tr>
<tr>
<td>a</td>
<td>b b (an @ expr.) c c</td>
</tr>
<tr>
<td>a</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>With L, R and C columns, and a <code>\multicolumn</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>the rain in spain the rain in spain falls mainly on the plain the rain in spain falls mainly on the plain</td>
</tr>
<tr>
<td>x</td>
<td>and now for something completely different</td>
</tr>
<tr>
<td>a</td>
<td>some multicolumn text across columns 2–4</td>
</tr>
<tr>
<td>a</td>
<td>b c d</td>
</tr>
<tr>
<td>a</td>
<td>b b c c d d</td>
</tr>
<tr>
<td>a</td>
<td></td>
</tr>
</tbody>
</table>

3
The following examples attempt to show the effect of the \texttt{\textbackslash tymin} and \texttt{\textbackslash tymax} parameters. One should also perhaps note that \texttt{\textbackslash tymax} refers to the total column width (including any inter-column space, rules etc) but \texttt{\textbackslash tymin} just refers to the width of the column entry (like the argument to the standard \texttt{p} column).

\texttt{\textbackslash tymin=0pt}  
\texttt{\textbackslash tymax=\textbackslash maxdimen}  
Note how the first column is ‘squeezed’. In fact it is in such a narrow column that even \texttt{‘a’} produces an overfull box warning!

\texttt{\textbackslash tymin=20pt}  
\texttt{\textbackslash tymax=\textbackslash maxdimen}  
Here increase \texttt{\textbackslash tymin} so that columns b and a are not so narrow. ‘a’ is set to its natural width, and ‘b’ is set to \texttt{\textbackslash tymin}.

\texttt{\textbackslash tymin=20pt}  
\texttt{\textbackslash tymax=200pt}  
In the previous example, the large d column dominated the table, being a lot wider than the c column. By reducing \texttt{\textbackslash tymax} can limit the width of column d producing more even column widths, but now producing an entry for d that is longer than that for c.
5 The Code

1 (*+package)
2 \RequirePackage{array}
3 \catcode`\Z=14
4 \DeclareOption{debugshow}{\catcode`\Z=9\relax}
5 \ProcessOptions

\arraybackslash Borrowed from tabularx.
6 \def\arraybackslash{\let\\=\@arraycr}

\@finalstrut Bug fixed version from December 1995 \LaTeX release. Old bug going back to \LaTeX2.09...
7 \def\@finalstrut#1{%
8 \unskip\ifhmode\nobreak\fi\vrule\@width\z@\@height\z@\@depth\dp#1}

\TY@count Counter so that we know what column we are hacking around in.
9 \newcount\TY@count

\tabulary Top level macro for standard form.
10 \def\tabulary{
11 \let\TY@final\tabular
12 \let\endTY@final\endtabular
13 \TY@tabular}

\TY@tabular Looks a lot like tabularx at this stage. Grab everything into a token register.
14 \def\TY@tabular#1{%
15 \edef\TY@{\@currenvir}\%
16 \{\ifnum0='\fi}
17 \@ovxx\TY@linewidth
18 \@ovyy\TY@tablewidth
19 \count@\z@
20 \@tempswatrue
21 \@whilesw\if@tempswa\fi{%
22 \advance\count@\@ne
23 \expandafter\ifx\csname TY@F\the\count@\endcsname\relax
24 \@tempswafalse
25 \else
26 \expandafter\let\csname TY@SF\the\count@\endcsname\relax
27 \csname TY@F\the\count@\endcsname
28 \global\expandafter\let\csname TY@F\the\count@\endcsname\relax
29 \expandafter\let\csname TY@S\the\count@\endcsname\relax
30 \expandafter\let\csname TY@\the\count@\endcsname\relax
31 \fi}%
Placing this here means that nested tabulars will get this definition but that’s probably OK, the extra code for LCR etc shouldn’t do any harm.

\TY@mkpream

TY version.

\def\TY@mkpream{%
\def\@addamp{%
  \if@firstamp \@firstampfalse \else
    \global\advance\TY@count\@ne
    \edef\@preamble{\@preamble &}
  \fi
\}
\def\@acol{%
  \TY@subwidth\col@sep
  \@addtopreamble{\hskip\col@sep}
\}
\let\@arrayrule\TY@arrayrule
\let\@classvi\TY@classvi
\def\@classv{
  \save@decl
  \expandafter\NC@ecs\@nextchar\extracolsep{\extracolsep\@@@}
  \sbox\z@{\dollarbegin\@nextchar\dollarend}
  \TY@subwidth{\wd\z@}
  \@addtopreamble{\dollarbegin\the@toks\the\count0\relax\dollarend}
  \prepnext@tok
}
\global\let\@mkpream\TY@@mkpream
\TY@@mkpream}

\TY@arrayrule

Pull this out so the colortbl support below can redefine

\def\TY@arrayrule{%
\TY@subwidth\arrayrulewidth
\@addtopreamble \vline}
\texttt{\@classvii}  Pull this out so the colortbl support below can redefine  
\begin{verbatim}
69 \def\@classvii{\ifcase \@lastchclass
70 \@acol \or
71 \@subwidth\doublerulesep
72 \@addtopreamble{\hskip \doublerulesep}\or
73 \@acol \or
74 \@classvi
75 \fi}
\end{verbatim}

\texttt{\@tab} First run a tabular with all the column types fudged so that the widths of any rules or \texttt{@}-expressions are noted.  
\begin{verbatim}
76 \def\@tab{\%
77 \setbox\z@\hbox\bgroup
Support displaymath by making it non-display in the first pass. (Other display environments defined in terms of $$ would need to be added here by packages that define them.)
78 \let\[\let\]%
79 \let\equation\let\endequation%
80 \col@sep\tabcolsep
81 \let\dollarbegin\begingroup\let\dollarend\endgroup
82 \let\@mkpream\@mkpream
83 \def\multicolumn##1##2##3{\multispan##1\relax}%%
84 \CT@start\@tabarray}
\end{verbatim}

\texttt{\@tabarray}  
\begin{verbatim}
85 \def\@tabarray{\\ifnextchar[{{\@array}{\@array[t]}}
86 \def\@array[#1]{\@array[t]}
\end{verbatim}

\texttt{\@width} Just a shorthand to access a column width macro.  
\begin{verbatim}
87 \def\@width#1{\expandafter#1\csname \@\the\@count\endcsname}
88 \end{verbatim}

\texttt{\@subwidth} Subtract a width from the current column width and also The total line table width and the target line width.  
\begin{verbatim}
89 \def\@subwidth#1{\@width\dimen@\advance\dimen@-#1\relax\@width\xdef{\the\dimen@}\global\advance\@linewidth-#1\relax}
\end{verbatim}

\texttt{\endtabulary} First run one modified tabular, making sure to add a blank row (cf longtable) to the end in case the user supplied last row is hidden by an hline or something.  
\begin{verbatim}
94 \def\endtabulary{\%
95 \gdef\@halignto{}
96 \expandafter\@tab\the\toks@
97 \crr\omit
\end{verbatim}
Check that $\text{tymin}$ is not too large.

Now take the last row apart, cf. longtable or appendix D.
A bit cheap just doing this four times, but prevents any possibilities of loop-
ing....

Reset the counter.

Reset the LCRJ column definition to set paragraphs to the calculated widths.

Run a second tabular, and for the star form, unbox it.

Finish off by restoring global commands.

\newcommand{\tymin}%
\global\advance\TY@tablewidth\dimen@%  
\global\expandafter\let\csname TY@F\the\TY@count\endcsname
\maxdimen
\if\fi
\advance\TY@count\m@ne
\repeat}%

\TY@checkmin
\TY@checkmin
\TY@checkmin
\TY@checkmin

\let\TY@box\TY@box@v
\expandafter\TY@final\the\toks@\endTY@final}%

\count@\z@
\@tempswatrue
\@whilesw\if@tempswa\fi{%
\advance\count@\@ne
\expandafter\ifx\csname TY@SF\the\count@\endcsname\relax
\@tempswafalse
\else
\global\expandafter\let\csname TY@F\the\count@\expandafter\endcsname
\csname TY@SF\the\count@\endcsname
\global\expandafter\let\csname TY@\the\count@\expandafter\endcsname
\csname TY@S\the\count@\endcsname
\fi}%
\TY@linewidth\@ovxx
\TY@tablewidth\@ovyy
\ifnum0='{\fi}

\TY@checkmin Check that no column is squeezed below \tymin. If it is, fix the width of that
column to \tymin and try again re-computing the ratio. (The new ratio will be
smaller, and may squeeze yet more rows, so need to iterate this, currently just do
it four times.)
\def\TY@checkmin{%
\let\TY@checkmin\relax
\ifdim\TY@tablewidth>\z@% 
\gscale@div\TY@ratio\TY@linewidth\TY@tablewidth
% \changes{v0.9}{2008/12/01}
\% \{cs\{TY@linewidth\}\}
\ifdim\TY@tablewidth < \TY@linewidth
\def\TY@ratio{1}\%
\else
\TY@warn{No suitable columns!}\%
\def\TY@ratio{1}\%
\fi
\count@\z@
\message{^^JLine Width: \the\TY@linewidth,}
\Z Natural Width: \the\TY@tablewidth,\Z
\TY@ratio``J}\
\@tempdima\z@
\loop
\ifnum\count@<\TY@count
\advance\count@\@ne
\ifdim\csname TY@F\the\count@\endcsname>\tymin
\dimen@\csname TY@	he\count@\endcsname
\dimen@\TY@ratio\dimen@
\ifdim\dimen@<\tymin
\Z \message{Column \the\count@ ->}\
\global\expandafter\let\csname TY@F\the\count@\endcsname\tymin
\global\advance\TY@linewidth-\tymin
\global\advance\TY@tablewidth-\csname TY@	he\count@\endcsname
\let\TY@checkmin\TY@@checkmin
\else
\expandafter\xdef\csname TY@F\the\count@\endcsname{\the\dimen@}\
\advance\@tempdima\csname TY@F\the\count@\endcsname
\fi
\fi
\Z \dimen@\csname TY@F\the\count@\endcsname\message{\the\dimen@, }\Z
\repeat
\message{^^JTotal: \the\@tempdima^^J}\
\TY@@checkmin Saved value
\let\TY@@checkmin\TY@checkmin
\TY@linewidth Stores the target width.
\newdimen\TY@linewidth
\tyformat What to do with columns
\def\tyformat{\everypar{\nobreak\hspace{z@skip}}}\
tymin Columns narrower than this are not fudged.
\newdimen\tymin
\tymin=10pt
Columns wider than this are all treated alike and set to the same width, to stop
one particularly long entry hijacking the entire table.
\newdimen\tymax
\tymax=2\textwidth

\@testpatch
Also add LCRJ although these don’t do anything useful except in tabulary.
\def\@testpatch{%
\ifnum \@lastchclass=6 \@ne \@chnum \@ne \else
\ifnum \@lastchclass=7 5 \else
\ifnum \@lastchclass=8 \tw@ \else
\ifnum \@lastchclass=9 \thr@@
\else \z@
\ifnum \@lastchclass = 10 \else
\edef\@nextchar{\expandafter\string\@nextchar}%
\@chnum
\if \@nextchar c\z@ \else
\if \@nextchar l\@ne \else
\if \@nextchar r\tw@ \else
\if \@nextchar s6 \else
\if \@nextchar C7 \else
\if \@nextchar L8 \else
\if \@nextchar R9 \else
\if \@nextchar J10 \else
\z@ \@chclass
\if\@nextchar |\@ne \else
\if \@nextchar !6 \else
\if \@nextchar @7 \else
\if \@nextchar <8 \else
\if \@nextchar >9 \else
10
\@chnum
\if \@nextchar m\thr@@\@else
\if \@nextchar p\thr@@ \else
\if \@nextchar b5 \else
\z@ \@chclass \z@ \@preamerr \z@ \fi \fi \fi \fi \fi \fi \fi \fi \fi
\fi
\fi \fi \fi \fi \fi \fi \fi \fi \fi \fi
\fi
\fi \fi
\fi
\fi
\fi
\fi
%}
\TY@classz
Here hacked around without the respect Frank’s code deserves...
\def\TY@class{%
\TY@box\centering The argument is \centering etc depending on whether LCRJ is used. However in this version the entries are set in horizontal mode with definitions mimicing the standard lcr columns. Later \TY@box will be redefined to \TY@box@v which really sets the entries in vertical mode.
\def\TY@box{\centering }
\ifx\centering\def\TY@box{\centering \kern\z@}
\fi
\def\TY@box@v{\vtop \@startpbox{\@nextchar}\insert@column \@endpbox}
\TY@tablewidth The natural width of the table on the first run.
\newdimen\TY@tablewidth
\Gscale@div Stolen from graphics package.
300 \def\Gscale@div\#1\#2\#3{% 
301 \setlength\dimen0\#3\%
302 \ifdim\dimen0=\z@ \PackageError{graphics}{Division by 0}\@eha 
303 \dimen0\#2\%
304 \fi
305 \edef\@tempd\the\dimen0%
306 \setlength\dimen0\#2\%
307 \count@65536\relax
308 \ifdim\dimen0<\z@ \dimen0-\dimen0 
309 \count@-\count@
310 \repeat
311 \divide\dimen0\dimen0 
312 \edef#1{strip@pt\dimen0}

\TY@get@body Place all tokens as far as the first \end into a token register. Then call \TY@find@end to see if we are at \end{tabulary}.
322 \long\def\TY@get@body#1\end{ 
323 \toks@\expandafter{\the\toks@#1}\expandafter\@tempa

\TY@find@end If we are at \end{tabulary}, call \end{tabulary}, otherwise add \end{...} to the register, and call \TY@get@body again.
324 \def\TY@find@end{}%
325 \def\@tempa{}%
326 \ifx\@tempa\TY@\def\@tempa{\end{\@tempa}}\expandafter\@tempa
327 \else\toks0\expandafter
328 \{\the\toks0\end{\@tempa}\expandafter\TY@get@body\fi

\TY@warn Warning messages.
329 \def\TY@warn{}
330 \PackageWarning{tabulary}}
331 \catcode´Z=11
332 \AtBeginDocument{
333 \ifpackageloaded{colortbl}{%
334 \expandafter\def\expandafter\@mkpream\expandafter{\expandafter#1\
335 \expandafter\let\expandafter\CT@setup\expandafter\relax
336 }%
\endgroup
\@tempdima\ht\z@
\advance\@tempdima\minrovclearance
\vrule\@height\@tempdima\@width\z@
\unhbox\z@
\prepnext@tok
\def\TY@arrayrule{\TY@subwidth\arrayrulewidth
\@addtopreamble{{\CT@arc@\vline}}}%
\def\TY@classvi{\ifcase \@lastchclass
\@acol \or
\TY@subwidth\doublerulesep
\if\CT@drsc@\relax
\@addtopreamble{{\CT@drsc@\vrule\@width\doublerulesep}}%
\else
\@addtopreamble{{\CT@drsc@\vrule\@width\doublerulesep}}%
\fi}\or
\@acol \or
\@classvii
\fi}%
{%
\let\CT@start\relax%
}{}
end of at begin document

\TX@warn \verb support, uses same csnames as in TX so they share code if both loaded (this version names tabulary in the warning though). See tabularx for documentation.

{\uccode'*=\ %
\uppercase{\gdef\TX@verb{\leavevmode\null\TX@vwarn
{\ifnum0='}\fi\ttfamily\let\ignorespaces
\afterassignment\TX@vfirst\let\@tempa=.\fi}}
\def\TX@v#1!{\afterassignment\TX@vfirst\let\@tempa=}
\begingroup
\catcode'*=\catcode'#=12
\gdef\TX@vfirst{\if\@tempa#%
\def\@tempa#1 *2{\fi}
\else
\let\@tempb\TX@v#
\if\@tempa\space~\else\@tempa\fi
\fi
\@tempb}
\gdef\TX@v#1 *2{\fi
Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.

Symbols
\# ........................... 420, 421
\* ........................... 411, 420
@@@ .................................. 59
\@acol .......................... 53, 70, 73, 397, 404
\addamp .......................... 48
\addtopreamble ........................ 55, 62, 68, 72, 269, 363, 395, 400, 402
\@array ........................ 85, 86
\@arraycr ........................ 6
\@arrayrule ........................ 56
\@chclass 221, 238, 249
\@chnum 222, 229, 245, 263, 266, 270, 357, 360, 365
\@classv ........................ 58
\@classvi ........................ 57
\@classvii 74, 405
\@classx ........................ 253, 346
\@classz ........................ 43
\@currenvir ........................ 15
\@depth ........................ 8
\@empty ........................ 303
\@firstampfalse 49
\@firststampfalse 49
\@halignto 95
\@height 8, 390
\@nil 356
\@ovxx 17, 172
\@ovyy 18, 173
\@preamble 51
\@preamerr 429
\@spaces 127
\@startpbox 275–277, 296, 371–373
\@tempx 325, 326, 416–418, 423, 427
\@tempa 325, 326, 416–418, 423, 427
\@tempb 424, 426, 429
\@tempcnta 306, 318
\@tempdim a 275–277, 298, 371–373
\@tempcnta 254, 347, 356
\@tempsvsize 24, 165
\@tempsvmargin 20, 161
\@tempsvspace 221
\@ifpackageloaded 333
\@ifwarning 435
\@ifstar 415
\@ifnextchar 85
\@whilesw 21, 162
A
\@warning 435
\@ifstar 415
B
\@warning 435
\@ifstar 415
C
\@warning 435
\@ifstar 415
Change History

v0.1
General: Initial version

v0.10
General: support \cellcolor see http://tex.stackexchange.com/a/185851/1090

v0.2
General: Changed everything except the name

v0.3
General: Changed everything except the name: s and CLRS

v0.4
\checkmin: \global added

v0.5
\checkmin

v0.6
General: Further SPQR modifications to multi pass table env

v0.7
\checkmin

v0.10
General: support \cellcolor see http://tex.stackexchange.com/a/185851/1090

v0.5
\checkmin

v0.6
General: Remove multi pass table env and unboxed star form

v0.7
\checkmin: new macro to support [t] optional arg

\arrayrule
\array
\box
\boxv
\checkmin
\classvi
\classz
\count
\final
\findend
\getbody
\linewidth
\mkpream
\ratio
\saverow
\subwidth
\tab
\tabarray
\tablewidth
\tabular
\warn
\width
\yformat
\ymin
\ymax
\uccode
\unBox
\unskip
\unVBox
\uppercase
\vbox
\vcenter
\verb
\vline
\vrule
\vtop
\wd
\xdef
\z@
\z@@
\z@skip
\z@
v0.8

General: Rename S to J and 'hide' s (until it works) . . . . . . . . . . . 1