The package \texttt{decision-table}* 

S. Vandevelde, F. Pantigny  
s.vandevelde@kuleuven.be  
fpantigny@wanadoo.fr  
December 8, 2020

\section*{Abstract}

The \LaTeX{} package \texttt{decision-table} provides a command \texttt{dmntable}, which allows for an easy way to generate decision tables in the Decision Model and Notation (DMN) format. (See Fig. 1) This package ensures consistency in the tables (i.e. fontsize), and is thus a better alternative to inserting tables via images.

\section{Description}

The \texttt{decision-table} package allows for an easy way to generate decision tables in the Decision Model and Notation (DMN) format. (See Fig. 1) This package ensures consistency in the tables (i.e. fontsize), and is thus a better alternative to inserting tables via images.

\begin{figure}[h]
\centering
\begin{tabular}{|l|c|c|}
\hline
\textbf{Calculate BMI} & \textbf{Weight(kgs)} & \textbf{Length(m)} & \textbf{BMI} \\
\hline
1 & - & - & \text{weight/length*length} \\
\hline
\end{tabular}
\caption{Example of a DMN table}
\end{figure}

\section{How to use}

The \texttt{decision-table} package adds the \texttt{dmntable} command, with which tables can be created. This command expands into a \texttt{tabular}, so it can be used within a \texttt{table} or \texttt{figure} environment. Furthermore, this allows labels and captions to be added seamlessly. It is also possible to place multiple DMN tables in one \texttt{table/figure} environment. The \texttt{dmntable} command has the following inputs:

- title
- hit policy
- input column headers
- output column headers
- the table values

The command is used as follows:

\begin{verbatim}
\% \texttt{dmntable \{title\}\{hitpolicy\}\{input\}\{output\}\{values\}}\%
\end{verbatim}

*This document corresponds to the version 0.0.3 of \texttt{decision-table}, at the date of 2020/12/08.
The input, output and cell values are split by a comma. It is not necessary to include the row numbers for the cell values. For example, 1 is generated by the following code:

```latex
\begin{figure}[H]
\centering
\dmntable{Calculate BMI}{U}{\begin{itemize}
\item Weight (kgs), Length (m) \{BMI\}
\end{itemize}}{\begin{itemize}
\item weight/length*length
\end{itemize}}{\caption{Example of a DMN table}}{\label{ex1}}
\end{figure}
```

If a cell value contains multiple values (e.g. multiple string values), then accolades should be written around them. See the example 4.

### 3 Examples

This section contains some example code and their resulting tables.

```latex
\begin{figure}[H]
\centering
\dmntable{Calculate BMI}{U}{\begin{itemize}
\item Weight (kgs), Length (m) \{BMI\}
\end{itemize}}{\begin{itemize}
\item weight/length*length
\end{itemize}}{\caption{Example of a DMN table}}
\end{figure}
```

![Figure 2: Example of a DMN table](image1.png)

```latex
\begin{figure}[H]
\centering
\dmntable{Decide BMI Level}{U}{\begin{itemize}
\item BMI \{BMI Level, Risk Level\}
\end{itemize}}{\begin{itemize}
\item \$< 18.5\$, Underweight, Increased,
\item \$[18.5..24.9]\$, Normal, Low,
\item \$[25..29.9]\$, overweight, Increased,
\item \$[30..34.9]\$, Obese I, High,
\item \$[35..39.9]\$, Obese II, Very High,
\item \$> 39.9\$, Extreme Obesity, Extremely High
\end{itemize}}{\caption{Example of a DMN table}}
\end{figure}
```

![Figure 3: Example of a larger DMN table](image2.png)
### Decide Risk Level

<table>
<thead>
<tr>
<th>U</th>
<th>BMI level</th>
<th>Waist(cm)</th>
<th>Risk Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Overweight</td>
<td>$\leq 88$</td>
<td>Increased</td>
</tr>
<tr>
<td>2</td>
<td>Overweight</td>
<td>$&gt; 88$</td>
<td>High</td>
</tr>
<tr>
<td>3</td>
<td>Obese I</td>
<td>$\leq 88$</td>
<td>High</td>
</tr>
<tr>
<td>4</td>
<td>Obese II</td>
<td>—</td>
<td>Very High</td>
</tr>
<tr>
<td>5</td>
<td>Extreme Obesity</td>
<td>—</td>
<td>Extremely High</td>
</tr>
<tr>
<td>6</td>
<td>not(Overweight, Obese I, Obese II, Extreme Obesity)</td>
<td>—</td>
<td>Low</td>
</tr>
</tbody>
</table>

Figure 4: Example of cell with multiple values

### Rule 1

<table>
<thead>
<tr>
<th>U</th>
<th>Age</th>
<th>Service Years</th>
<th>Eligible1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$\leq 18$</td>
<td>—</td>
<td>Yes</td>
</tr>
<tr>
<td>2</td>
<td>$\geq 60$</td>
<td>—</td>
<td>Yes</td>
</tr>
<tr>
<td>3</td>
<td>$[18, 60]$</td>
<td>$\geq 30$</td>
<td>Yes</td>
</tr>
</tbody>
</table>

### Rule 2

<table>
<thead>
<tr>
<th>U</th>
<th>Age</th>
<th>Service Years</th>
<th>Eligible2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>—</td>
<td>$\geq 30$</td>
<td>Yes</td>
</tr>
<tr>
<td>2</td>
<td>$\geq 60$</td>
<td>$&lt; 30$</td>
<td>Yes</td>
</tr>
</tbody>
</table>

### Rule 3

<table>
<thead>
<tr>
<th>U</th>
<th>Age</th>
<th>Service Years</th>
<th>Eligible3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$&lt; 45$</td>
<td>$[15, 30)$</td>
<td>Yes</td>
</tr>
<tr>
<td>2</td>
<td>$&gt; 45$</td>
<td>—</td>
<td>Yes</td>
</tr>
</tbody>
</table>

### Vacation Days

<table>
<thead>
<tr>
<th>C+</th>
<th>Eligible1</th>
<th>Eligible2</th>
<th>Eligible3</th>
<th>Vacation Days</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>22</td>
</tr>
<tr>
<td>2</td>
<td>Yes</td>
<td>—</td>
<td>—</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>—</td>
<td>Yes</td>
<td>—</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>No</td>
<td>—</td>
<td>Yes</td>
<td>2</td>
</tr>
</tbody>
</table>

Figure 5: DMN table example for the Vacation Days

## 4 Contributing

Contributions are always welcome. The project is hosted at [https://gitlab.com/Vadevesi/dmn-tex](https://gitlab.com/Vadevesi/dmn-tex)
5 Implementation

We give the traditional declaration of a package written with expl3:

\RequirePackage{l3keys2e}
\ProvidesExplPackage\{decision-table\} \{\dmnfiledate\}
\ProvidesExplPackage\{\dmnfileversion\}
\ProvidesExplPackage\{Table of decision\}
\RequirePackage{nicematrix}

We define the command \dmntable with the tools of xparse (in 2020 October, xparse will be included in the LaTeX kernel).

\NewDocumentCommand\dmntable{m m m m m}{
\clist_clear_new:N \l_@@_input_clist
\clist_set:Nn \l_@@_input_clist { #3 }
The clist (command separated list) \l_@@_input_clist is for the list of the names of the input fields.
\clist_clear_new:N \l_@@_output_clist
\clist_set:Nn \l_@@_output_clist { #4 }
The clist \l_@@_output_clist is for the list of the names of the output fields.
\int_zero_new:N \l_@@_input_int
\int_set:Nn \l_@@_input_int { \clist_count:N \l_@@_input_clist }
The integer \l_@@_input_int is the number of the input fields.
\int_zero_new:N \l_@@_output_int
\int_set:Nn \l_@@_output_int { \clist_count:N \l_@@_output_clist }
The integer \l_@@_output_int is the number of the output fields.
\seq_clear_new:N \l_@@_cells_seq
\seq_set_split:Nnn \l_@@_cells_seq { , } { #5 }
The sequence \l_@@_cells_seq is the sequence of all the cells of the “body” of the tabular.
\use:x
\begin{NiceTabular}
\use:x\begin{NiceTabular}
Here is the preamble of the tabular. The command \texttt{\prg\replicate:nn} is expandable and hence will be expanded by the \texttt{use:x}.

\begin{verbatim}
\{ \prg\replicate:nn \{ \l_@@_input_int + \l_@@_output_int \} 1 \}
\end{verbatim}

Here is the list of options of the \texttt{\{NiceTabular\}} (a standard tabular of \texttt{\{array\}} don’t have such list of options. Once again, we have to compute some quantities in this list of options before the execution of \texttt{\begin{NiceTabular}}.

\begin{verbatim}
[ \texttt{hvlines-except-corners} \]
\end{verbatim}

The key \texttt{hvlines-except-corners} will draw all the rules of the tabular, excepted in the (upper right) corner.

\texttt{hvlines-except-corners},

The key \texttt{code-before} of \texttt{\{NiceTabular\}} contains instructions to color the cells \texttt{before} the rules (doing so, the resulting PDF gives better results in the PDF viewers).

\texttt{code-before =}

First, a \texttt{\rectanglecolor} for the labels of the “input” fields. The command \texttt{\int_eval:n} is expandable and, hence, will be expanded by the \texttt{use:x}. On the other side, we have to prevent the expansion of \texttt{\rectanglecolor} which, in fact, at that point is not defined (it will be defined by \texttt{nicematrix} after the construction of the array).

\begin{verbatim}
\exp_not:N \rectanglecolor
\{ blue!10!green!60!black!30 \}
\{ 2 - 2 \}
\{ 2 - \int_eval:n \{ \l_@@_input_int + 1 \} \}
\end{verbatim}

A \texttt{\rectanglecolor} for the labels of the “output” fields.

\begin{verbatim}
\exp_not:N \rectanglecolor
\{ green!30!blue!15 \}
\{ 2 - \int_eval:n \{ \l_@@_input_int + 2 \} \}
\{ 2 - \int_eval:n \{ \l_@@_input_int + \l_@@_output_int + 1 \} \}
\end{verbatim}

Now, we begin the body of the tabular (the environment \texttt{\{NiceTabular\}}).

The body begins by a \texttt{\multicolumn} for the title. However, we have to compute the number of cells of that \texttt{\multicolumn}. That’s why we have to expand the first argument of the \texttt{\multicolumn} before executing the \texttt{\multicolumn}. However, we have to do that in an expandable way in order to prevent the functionality of the \texttt{\multicolumn} (which internally give a \texttt{\omit} of TeX). That’s why we have to use \texttt{\exp_args:Ne} (\texttt{\exp_args:Nx} would not do the job).

\begin{verbatim}
\exp_args:Ne \multicolumn
\{ \int_eval:n \{ \l_@@_input_int + 1 \} \}
\{ 1 \}
\{ #1 \}\}
\end{verbatim}

\#2 is the \texttt{hit policy}.

\begin{verbatim}
#2 &
\end{verbatim}

Now, the fields (“input fields” and “output fields”). By using \texttt{\clist_use:Nn}, we replace the commas by ampersands (\&).

\begin{verbatim}
\clist_use:Nn \l_@@_input_clist { & } &
\clist_use:Nn \l_@@_output_clist { & } \\}
\end{verbatim}
Now, all the rows corresponding to the rules. We begin a loop over all the cells with `seq_map_inline:Nn`.

\begin{NiceTabular}
\seq_map_inline:Nn \l_@@_cells_seq
\c@jCol \text{and} \c@iRow \text{are counters provided by \texttt{NiceTabular} for the current column and the current row. If you are in the first column, we insert the number of rule.}
\int_compare:nT { \c@jCol = 0 } { \int_eval:n { \c@iRow - 1 } \& }\end{NiceTabular}

Now, we add one composante of \l_@@_cells_seq.

Before the following cell, we have, of course, to add \textbackslash (if we are at the end of the row) or \& (elsewhere).
\begin{NiceTabular}
\int_compare:nTF { \c@jCol = \l_@@_input_int + \l_@@_output_int + 1 }
{ \textbackslash \& }
{ \& }
\end{NiceTabular}

\textbf{Contents}

1 Description 1
2 How to use 1
3 Examples 2
4 Contributing 3
5 Implementation 4