

# Using table1 with L<sup>A</sup>T<sub>E</sub>X and MS Word

Benjamin Rich

2025-09-19

## Introduction

A long requested feature has been the ability to use `table1` in `rmarkdown` documents that render to L<sup>A</sup>T<sub>E</sub>X or `.docx` (i.e. Microsoft<sup>®</sup> Word). Since version 1.4 of `table1`, this is now possible (with some limitations) by converting the output of `table1()` to a `data.frame`, `kableExtra` or `flextable`, using the functions `as.data.frame()`, `tikable()` and `tiflex()` respectively, as these objects can be rendered to L<sup>A</sup>T<sub>E</sub>X (note: `data.frame` (via `pandoc`) and `flextable` objects can also be rendered to `.docx` format, while, `kableExtra` cannot).

## Examples

We demonstrate this using a familiar example from the main vignette. First, we can try the `data.frame` approach:

```
x <- table1(~ age + sex + wt | treat, data=dat)
as.data.frame(x)
```

		Placebo	Treated	Overall
		(N=52)	(N=94)	(N=146)
## 1				
## 2	Age (years)			
## 3	Mean (SD)	39.2 (14.2)	40.1 (13.3)	39.8 (13.6)
## 4	Median [Min, Max]	37.5 [18.0, 65.0]	39.5 [18.0, 65.0]	39.0 [18.0, 65.0]
## 5	Sex			
## 6	Female	34 (65.4%)	53 (56.4%)	87 (59.6%)
## 7	Male	18 (34.6%)	41 (43.6%)	59 (40.4%)
## 8	Weight (kg)			
## 9	Mean (SD)	68.1 (16.3)	68.3 (16.7)	68.2 (16.5)
## 10	Median [Min, Max]	66.7 [37.5, 116]	64.9 [40.0, 119]	66.2 [37.5, 119]
## 11	Missing	2 (3.8%)	3 (3.2%)	5 (3.4%)

By default, this does not produce a L<sup>A</sup>T<sub>E</sub>X table, but the same text output you would see in the R console. If the `printr` package is loaded, however, we do get a L<sup>A</sup>T<sub>E</sub>X table by default:

```
library(printr, quietly=TRUE)
```

```
## Registered S3 method overwritten by 'printr':
##   method      from
##   knitr_print.data.frame rmarkdown
```

```
as.data.frame(x)
```

	Placebo	Treated	Overall
	(N=52)	(N=94)	(N=146)
Age (years)			
Mean (SD)	39.2 (14.2)	40.1 (13.3)	39.8 (13.6)
Median [Min, Max]	37.5 [18.0, 65.0]	39.5 [18.0, 65.0]	39.0 [18.0, 65.0]
Sex			
Female	34 (65.4%)	53 (56.4%)	87 (59.6%)
Male	18 (34.6%)	41 (43.6%)	59 (40.4%)
Weight (kg)			
Mean (SD)	68.1 (16.3)	68.3 (16.7)	68.2 (16.5)
Median [Min, Max]	66.7 [37.5, 116]	64.9 [40.0, 119]	66.2 [37.5, 119]
Missing	2 (3.8%)	3 (3.2%)	5 (3.4%)

Alternatively, we can use the `knitr::kable()` function:

```
kable(as.data.frame(x), booktabs=TRUE)
```

	Placebo	Treated	Overall
	(N=52)	(N=94)	(N=146)
Age (years)			
Mean (SD)	39.2 (14.2)	40.1 (13.3)	39.8 (13.6)
Median [Min, Max]	37.5 [18.0, 65.0]	39.5 [18.0, 65.0]	39.0 [18.0, 65.0]
Sex			
Female	34 (65.4%)	53 (56.4%)	87 (59.6%)
Male	18 (34.6%)	41 (43.6%)	59 (40.4%)
Weight (kg)			
Mean (SD)	68.1 (16.3)	68.3 (16.7)	68.2 (16.5)
Median [Min, Max]	66.7 [37.5, 116]	64.9 [40.0, 119]	66.2 [37.5, 119]
Missing	2 (3.8%)	3 (3.2%)	5 (3.4%)

The output here is a bit nicer because we have specified the `booktabs` option, but because we are talking about a simple `data.frame`, there is no option to specify formatting (like bold text for variable labels, for instance).

Next, we can try the `tblkable()` function, to produce a `kableExtra` object:

```
tblkable(x)
```

	<b>Placebo (N=52)</b>	<b>Treated (N=94)</b>	<b>Overall (N=146)</b>
<b>Age</b> (years)			
Mean (SD)	39.2 (14.2)	40.1 (13.3)	39.8 (13.6)
Median [Min, Max]	37.5 [18.0, 65.0]	39.5 [18.0, 65.0]	39.0 [18.0, 65.0]
<b>Sex</b>			
Female	34 (65.4%)	53 (56.4%)	87 (59.6%)
Male	18 (34.6%)	41 (43.6%)	59 (40.4%)
<b>Weight</b> (kg)			
Mean (SD)	68.1 (16.3)	68.3 (16.7)	68.2 (16.5)
Median [Min, Max]	66.7 [37.5, 116]	64.9 [40.0, 119]	66.2 [37.5, 119]
Missing	2 (3.8%)	3 (3.2%)	5 (3.4%)

This looks better: it uses the `booktabs` option by default, variable labels and column headers are bold, and the alignment matches the HTML defaults (first column left-aligned, the remainder centered). Additional customizations can be done using function arguments or additional functions from the `kableExtra` package. One important thing to note is that the object is created using the `escape = FALSE` option from the `kable()` function. This has the advantage that HTML or  $\text{\LaTeX}$  code can be inserted directly into the output, but care must be taken to escape any special symbols manually or else the table won't look as expected, or it might even lead to an error on compilation of the  $\text{\LaTeX}$ . The one exception is the the percent symbol (“%”) will be escaped internally automatically.

Finally, we can try using the `tiflex()` function to produce a `flextable` object:

```
tiflex(x)
```

```
## Warning: fonts used in 'flextable' are ignored because the 'pdflatex' engine is used and not
## 'xelatex' or 'lualatex'. You can avoid this warning by using the
## 'set_flextable_defaults(fonts_ignore=TRUE)' command or use a compatible engine by defining
## 'latex_engine: xelatex' in the YAML header of the R Markdown document.
```

	<b>Placebo (N=52)</b>	<b>Treated (N=94)</b>	<b>Overall (N=146)</b>
<b>Age (years)</b>			
Mean (SD)	39.2 (14.2)	40.1 (13.3)	39.8 (13.6)
Median [Min, Max]	37.5 [18.0, 65.0]	39.5 [18.0, 65.0]	39.0 [18.0, 65.0]
<b>Sex</b>			
Female	34 (65.4%)	53 (56.4%)	87 (59.6%)
Male	18 (34.6%)	41 (43.6%)	59 (40.4%)
<b>Weight (kg)</b>			
Mean (SD)	68.1 (16.3)	68.3 (16.7)	68.2 (16.5)
Median [Min, Max]	66.7 [37.5, 116]	64.9 [40.0, 119]	66.2 [37.5, 119]
Missing	2 (3.8%)	3 (3.2%)	5 (3.4%)

(Note that that `flextable` output, in particular the font, is different depending on whether the `xelatex`, `lualatex` or `pdflatex` engine is used, and `flextable` emits a warning when `pdflatex`, the default for `rmarkdown` documents, is used because `xelatex` and `lualatex` gives more font options.)

The output is less attractive in my opinion, and less consistent with the typical look of L<sup>A</sup>T<sub>E</sub>X tables and documents. It is more spaced out, doesn't have bold labels or line break in the column headers (actually, this is a problem common to both packages, but the `tblkable()` and `tblflex()` functions deal with it differently because `flextable` will actually include the line breaks in HTML and .docx output, and the overall best approach is unclear), and doesn't use the `booktabs` package (the thickness of the horizontal rules is different). Also, `flextable` places the table in a float, whereas `kableExtra` leaves it inline. I personally would use `kableExtra` over `flextable` in a L<sup>A</sup>T<sub>E</sub>X documents, and in fact I have made this the default output in a L<sup>A</sup>T<sub>E</sub>X context (i.e., when using `rmarkdown` with a L<sup>A</sup>T<sub>E</sub>X output format). But a big advantage of `flextable` is that it can render to .docx format (i.e. Microsoft Word), which `kableExtra` cannot, and is the default output in that context.

## Nested groups

Nested groups are supported with `kableExtra` and `flextable`, but not simple `data.frames`. Here, one example with `kableExtra`:

```
x2 <- table1(~ age + wt | treat*sex, data=dat, overall=FALSE)
tblkable(x2)
```

	Placebo		Treated	
	Female (N=34)	Male (N=18)	Female (N=53)	Male (N=41)
<b>Age</b> <span>&lt;span class='varunits'&gt; (years)&lt;/span&gt;</span>				
Mean (SD)	40.6 (14.5)	36.6 (13.6)	40.1 (13.4)	40.1 (13.3)
Median [Min, Max]	39.5 [18.0, 65.0]	33.5 [18.0, 64.0]	39.0 [18.0, 65.0]	41.0 [18.0, 65.0]
<b>Weight</b> <span>&lt;span class='varunits'&gt; (kg)&lt;/span&gt;</span>				
Mean (SD)	68.8 (14.8)	66.8 (19.3)	65.6 (15.1)	71.5 (18.0)
Median [Min, Max]	67.2 [45.8, 116]	66.6 [37.5, 105]	61.4 [41.9, 103]	68.3 [40.0, 119]
Missing	1 (2.9%)	1 (5.6%)	3 (5.7%)	0 (0%)

## Captions and footnotes

Captions and footnotes are supported with both `kableExtra` and `flextable`. An example with `kableExtra`:

```
x <- table1(~ age + sex + wt | treat, data=dat,
  caption="Test caption", footnote="Test footnote")
tblkable(x)
```

Alternatively, the `kableExtra::footnote()` function can be used to add footnotes to the resulting object (this function also does automatic numbering, which may be considered an advantage). Note that when a caption is used, the table is rendered as a float.

## Closing remarks

As of version 1.4, it is now possible to use `table1` within L<sup>A</sup>T<sub>E</sub>X documents. There are some limitations on the formatting, however (it looks fine, but not quite the same as the HTML output and there is less flexibility to control it).

Table 4: Test caption

	Placebo (N=52)	Treated (N=94)	Overall (N=146)
<b>Age</b> <span class="varunits">&gt; (years)&lt;/span&gt;</span>			
Mean (SD)	39.2 (14.2)	40.1 (13.3)	39.8 (13.6)
Median [Min, Max]	37.5 [18.0, 65.0]	39.5 [18.0, 65.0]	39.0 [18.0, 65.0]
<b>Sex</b>			
Female	34 (65.4%)	53 (56.4%)	87 (59.6%)
Male	18 (34.6%)	41 (43.6%)	59 (40.4%)
<b>Weight</b> <span class="varunits">&gt; (kg)&lt;/span&gt;</span>			
Mean (SD)	68.1 (16.3)	68.3 (16.7)	68.2 (16.5)
Median [Min, Max]	66.7 [37.5, 116]	64.9 [40.0, 119]	66.2 [37.5, 119]
Missing	2 (3.8%)	3 (3.2%)	5 (3.4%)
Test footnote			

The actual  $\text{\LaTeX}$  code generation is handled by an external package (i.e., `knitr`, `kableExtra` or `flextable`). The  $\text{\LaTeX}$  generated by `kableExtra` and `flextable` is very different, and in my opinion `kableExtra` produces the better result. Both `kableExtra` and `flextable` have an extensive number of functions that can be applied to the converted object in order to alter the appearance of the table but these will not be described here; refer to each package's documentation for the complete details.