Best Practice for newline in LaTeX table

Hao

2024-01-23

Since many people have asked me this question (#157, this SO question, etc.), I feel like I should document it out. :)

Wrapping texts and make newlines may seem to be the same but they are actually quite different. Thinking about when you “wrap texts” in your text editor, you have a fixed width window and the texts will be automatically wrapped. It’s like a passive skill (in games :P). However, when you are trying to make newlines, you are inserting the linebreak by yourself and it’s mostly like an active skill you need to cast. For these two tasks, LaTeX provides two totally different approaches.

Text wrapping

If you are only trying stop your texts from “overflowing”, you can get it done by setting a fixed width with \texttt{kableExtra::column_spec}. This is the most recommended practice as it’s fairly straightforward. The column width controls the width for both table header and table body.

```r
library(kableExtra)

dt <- data.frame(
  Items = c("Item 1", "Item 2", "Item 3"),
  Text_1 = c("Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin vehicula tempor ex. Morbi malesuada",
  Text_2 = c("Duis posuere placerat magna, ac aliquam lorem viverra non. Ut ultrices tempus eros, quis auctor

kable(dt, "latex", booktabs = T,
  col.names = c("Item", "Short Title", "Very Very Very Very Very Very Long Title")) %>%
  column_spec(2:3, width = "5cm")
```
<table>
<thead>
<tr>
<th>Item</th>
<th>Short Title</th>
<th>Very Very Very Very Very Very Long Title</th>
</tr>
</thead>
</table>

Insert linebreak in table

In LaTeX, to make linebreaks in table cells, people usually use the makecell package. kableExtra 0.8.0 comes with a function called linebreak to facilitate that. Basically, this function will scan the existence of \n. If \n exists, it will put the texts in a makecell statement. It works in a very similar way with cell_spec so you will need to put escape = F in kable.

```latex
\text{linebreak}("a
b")
```

When you have \n in your data frame, you can either change the value manually or simply use it with mutate_all.

```r
dt2 <- data.frame(
  Item = c("Hello\nWorld", "This\nis a cat"),
  Value = c(10, 100)
)
dt2$Item <- linebreak(dt2$Item)
```

```latex
\begin{tabular}{|l|l|}
\hline
Item & Value \\
\hline
Hello\nWorld & 10 \\
This\nis a cat & 100 \\
\hline
\end{tabular}
```

Note that linebreaks in table captions should be treated in a different way. You will need to insert the linebreak (\\) by yourself manually, which is easier than using linebreak. linebreak doesn’t work because it put things in a makecell, which doesn’t work in caption (as it’s not a cell :P). At the same time, if you are using kableExtra 0.9.0 or any previous version, you need to load the caption package in LaTeX by yourself.
Table 1: Main Title
Subtitle

<table>
<thead>
<tr>
<th>Item (Name)</th>
<th>Value (Number)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hello</td>
<td>10</td>
</tr>
<tr>
<td>World</td>
<td>100</td>
</tr>
</tbody>
</table>

header-includes:
- \usepackage{caption}

Linebreak in other kableExtra functions

If you have a need to put a linebreak in kableExtra functions such as add_header_above and pack_rows, just go ahead and use \n directly (in kableExtra >= 0.8.0) and it will be automatically converted. Note that this feature is also controlled by the escape option in those functions.

```r
dt2 %>%
  kable("latex", booktabs = T, escape = F,
         col.names = linebreak(c("Item\n(Name)", "Value\n(Number)")), align = "c") %>%
  add_header_above(c("Combined\nTitle" = 2)) %>%
  pack_rows("Group\n1", 2, 2)
```

<table>
<thead>
<tr>
<th>Combined Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item (Name)</td>
</tr>
<tr>
<td>------------</td>
</tr>
<tr>
<td>Hello</td>
</tr>
<tr>
<td>World</td>
</tr>
</tbody>
</table>

**Group 1**
| This is a cat | 100 |