

Best Practice for newline in LaTeX table

Hao

2018-04-02

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 `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.

```
library(kableExtra); library(tidyverse)
dt <- tibble(
  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 sagittis turpis, at venenatis nisl luctus a.",
  Text_2 = c("Duis posuere placerat magna, ac aliquam lorem viverra non. Ut ultrices tempus eros, quis sodales libero commodo non. In non neque ut lacus vestibulum dictum a quis ipsum. Aenean ut justo interdum, laoreet enim nec, viverra eros. Donec vel pharetra nunc. Suspendisse vel ipsum ac lectus semper aliquam ac a orci. Suspendisse libero mauris, egetas semper auctor sit amet, tempor et orci. Phasellus quis neque aliquet, finibus nunc eget, lacinia neque. Sed auctor lectus vel ex scelerisque commodo.")
)

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")
```

Item	Short Title	Very Very Very Very Very Very Long Title
Item 1	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin vehicula tempor ex. Morbi malesuada sagittis turpis, at venenatis nisl luctus a.	Duis posuere placerat magna, ac aliquam lorem viverra non. Ut ultrices tempus eros, quis sodales libero commodo non. In non neque ut lacus vestibulum dictum a quis ipsum.
Item 2	In eu urna at magna luctus rhoncus quis in nisl. Fusce in velit varius, posuere risus et, cursus augue. Duis eleifend aliquam ante, a aliquet ex tincidunt in.	Aenean ut justo interdum, laoreet enim nec, viverra eros. Donec vel pharetra nunc. Suspendisse vel ipsum ac lectus semper aliquam ac a orci. Suspendisse libero mauris, egetas semper auctor sit amet, tempor et orci.
Item 3	Vivamus venenatis egetas eros ut tempus. Vivamus id est nisi. Aliquam molestie erat et sollicitudin venenatis. In ac lacus at velit scelerisque mattis.	Phasellus quis neque aliquet, finibus nunc eget, lacinia neque. Sed auctor lectus vel ex scelerisque commodo.

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 in a proper format. It works in a very similar way with `cell_spec`.

```
linebreak("a\nb")
```

```
## [1] "\\makecell[l]{a\\\\b}"
```

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

```
dt2 <- data.frame(  
  Item = c("Hello\nWorld", "This\nis a cat"),  
  Value = c(10, 100)  
)  
  
dt2 %>%  
  mutate_all(linebreak) %>%  
  kable("latex", booktabs = T, escape = F,  
        col.names = linebreak(c("Item\n(Name)", "Value\n(Number)"), align = "c"))
```

Item (Name)	Value (Number)
Hello World	10
This is a cat	100

Linebreak in other kableExtra functions

If you have a need to put a linebreak in `kableExtra` functions such as `add_header_above` and `group_rows`, just go ahead and use `\n` and it will be automatically converted. Note that this feature is controlled by the `escape` option in those functions.

```
dt2 %>%  
  mutate_all(linebreak) %>%  
  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)) %>%  
  group_rows("Group\n1", 2, 2)
```

Combined Title	
Item (Name)	Value (Number)
Hello World	10
Group 1	
This is a cat	100