

Number lines in code blocks with Rmarkdown

Atusy

Before reading

- For quick start, go “Autonumbering”.
- A source and output examples in variety of formats are on GitHub [atusy/rmd-line-num](#).
- Ask me questions on GitHub Issues or on Twitter (@Atsushi776)

Number Pandoc’s fenced code

Pandoc has an official support to number lines on fenced code by giving a class attribute `numberLines` (https://www.pandoc.org/MANUAL.html#extension-fenced_code_attributes).

Note that class attributes require `.` before the class name.

```
1 ```{r .numberLines}
2 x <- rnorm(10)
3 mean(x)
4 ```
```

Should become

```
1 x <- rnorm(10)
2 mean(x)
```

This is enough for `rmarkdown::pdf_document`.

For `html_document`, we also need `highlight:pygment` in YAML front matter¹.

```
1 ---
2 title: Line numbers with Rmarkdown documents
3 output:
4   html_document:
5     highlight: pygment
6   pdf_document: default
7 ---
```

Number Rmarkdown’s chunks and outputs

The above success infers a success in chunks of Rmarkdown.

¹“Chunk numberLines hook]” on Rpubs figured out requirement of `pygment` (<https://rpubs.com/Thell/numberLines>).

Numbering chunks

For code chunks of Rmarkdown documents, `numberLines` class can be given by assigning `class.source = "numberLines"` as a chunk option².

Again, you need `highlight: pygment` for `html_document`.

Thus,

```
1 ---
2 title: Line numbers with Rmarkdown documents
3 output:
4   html_document:
5     highlight: pygment
6 ---
7
8 ```{r, class.source = "numberLines"}
9 x <- seq(10)
10 mean(x)
11 ```
```

becomes

```
1 x <- seq(10)
2 mean(x)
```

```
## [1] 5.5
```

Great again!!

Unfortunately, line numbering does not work on `rmarkdown::word_document`.

It does not work properly on `rmarkdown::html_notebook` and `blogdown::html_page` as well. I guess some tricks required in CSS or JS.

Numbering outputs

You may also want to number lines on outputs by `class.output = "numberLines"`. However, this changes background colors to gray.

```
1 ## [1] 5.5
```

If output format is html, css will help.

Before adding `class.output = "numberLines"`, an output in html is

```
1 <pre><code>## [1] 5.5</code></pre>
```

After adding it, the output in html becomes

²See “Tex Results” section of “Chunk options and package options” by Yi Hui (<https://yihui.name/knitr/options/#text-results>)

```
1 <div class="sourceCode" id="cb7"><pre class="sourceCode numberSource numberLines"><code class="sourceCode
```

When `chunkout` class is further added, the output in html becomes

```
1 <div class="sourceCode" id="cb10"><pre class="sourceCode numberSource numberLines chunkout"><code class="
```

You can see `chunkout` class is added to `pre` tag.

So, lets modify css with

```
1 div.sourceCode pre.chunkout {  
2   background: white;  
3 }
```

and you'll be happy, right?

```
1 ## [1] 5.5
```

Sorry I do not support pdf because *LaTeX* kills me.

Autonumbering

- Super easy by a following template and edit after `<!-- Start your body -->`.
- Disable numbering by `class.source = NULL`, `class.output = NULL`.
- Note autonumbering does not work on Pandoc's fenced code ³.

```
1 ---  
2 output:  
3   html_document:  
4     highlight: pygment  
5     pandoc_args:  
6       - --indented-code-classes  
7       - lineNumbers  
8 ---
```

Input with a template

```
1 ---  
2 output:  
3   html_document:  
4     highlight: pygment  
5     pdf_document: default  
6 ---  
7  
8 ```{r setup, include = FALSE}
```

³The document indicates a following YAML front matter should work, but doesn't (<https://www.pandoc.org/MANUAL.html#reader-options>).

```

9 knitr::opts_chunk$set(
10   class.source = "numberLines",
11   class.output = c("numberLines", "chunkout")
12 )
13
14 # Add some arbitrary setup codes
15 ```
16
17
18 ```{css, echo = FALSE}
19 div.sourceCode pre.chunkout {
20   background: white;
21 }
22 ```
23
24 <!-- Start your body -->
25
26 **Numbered**
27
28 ```{r}
29 x <- seq(10)
30 mean(x)
31 ```
32
33 **Unnumbered**
34
35 ```{r, class.source = NULL, class.output = NULL}
36 x <- seq(10)
37 mean(x)
38 ```

```

Output from the template

Numbered

```

1 x <- seq(10)
2 mean(x)

```

```

1 ## [1] 5.5

```

Unnumbered

```

x <- seq(10)
mean(x)

```

```
## [1] 5.5
```

Enjoy !