Number lines in code blocks with Rmarkdown

Atusy

Go `atusy/rmd-line-num` on GitHub to see source files and output examples in variety of formats.

In pandoc style 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.

````
```{.r .numberLines}
x <- rnorm(10)
mean(x)
```
````

becomes

```r
x <- rnorm(10)
mean(x)
```

Great!

In Rmarkdown's chunk

The above success infers a success in chunks of Rmarkdown.

For code chunks of Rmarkdown documents, `numberLines` class can be given by assigning `class.source = "numberLines"` as a chunk option. [1]

This is enough for `rmarkdown::pdf_document`. On the other hand, `rmarkdown::html_document` requires `highlyght: pygment` in a YAML front matter[2].

Thus,

````
---
title: Line numbers with Rmarkdown documents
output:
  html_document:
    highlight: pygment
---

```{r, class.source = "numberLines"}
x <- seq(10)
mean(x)
```
````

[1] See "Tex Results" section of "Chunk options and package options" by Yi Hui (https://yihui.name/knitr/options/#text-results)

[2] "Chunk numberLines hook]" on Rpubs figured out requirement of `pygment` (https://rpubs.com/Thell/numberLines).

becomes

```
x <- seq(10)
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] 5.5
```

If output format is html, css will help.

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

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

After adding it, the output in html becomes

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

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

```
<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

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

and you'll be happy, right?

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

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

Numbering in default

Super easy.

Run following at just after a YAML front matter.

````
```{r setup}
knitr::opts_chunk$set(
  class.source = "numberLines",
  class.output = c("numberLines", "chunkout")
)
```

```{css}
div.sourceCode pre.chunkout {
  background: white;
}
```
````

Note that this does not affect pandoc style fenced code. According to pandoc's document, setting up a following YAML front matter should work, but doesn't.

```
output:
  html_document:
    highlight: pygment
      panoc_args:
        - --indented-code-classes
        - numberLines
```

Enjoy !