

可重复性研究

Reproducible Research

Mr. Black

目录

- Markdown
- R Markdown
- R Notebook

Markdown

Markdown

Markdown是一个由约翰（John Gruber）创建了一种轻量级标记语言。它允许人们使用易读易写的纯文本格式编写文档，然后转换成有效的XHTML（或者HTML）文档。本节内容主要参考了约翰对Markdown的介绍^[1]。

Markdown最大的特点是其易读性和易写性，Markdown文件可以直接以纯文本形式发布，而不会像HTML那样包含许多标签。Markdown的语法是有一些符号构成，其种类不是很多，仅对应了HTML标记中的一小部分。Markdown是一种书写格式，发布时程序会将Markdown文件转换成HTML格式文件。因此对于不在Markdown覆盖范围内的标签，可以直接将HTML标签写在Markdown文档中，同时不需要额外标注是HTML标签还是Markdown标签。

Markdown文件中，在HTML区块标签中的Markdown语法不会被处理。

[1] J. Gruber, “Markdown project.” <http://daringfireball.net/projects/markdown/>.

Markdown

- 段落和换行

一个段落通常是由一个或多个空行分割的一个或多个连续的文本行所组成。只要是看起来是空的行即可视为空行，例如：某一行包含了空格或制表符，但其仍被视为空行。普通段落不应该用空格或制表符进行缩进。

“由一个或多个连续的文本行所组成”这句话暗示了Markdown允许段落内部强制换行。和其他的文本转HTML格式不同，其他的格式会将每个换行符都转换成`
`标签。如果需要使用Markdown插入`
`时，需要在待插入位置输入两个以上的空格并按回车实现。尽管这样可能相对费事，但是把每个换行符都转换成`
`标签对于Markdown来说并不适合，因为Markdown在区块引用和多段落的列表中使用了换行来进行排版，这样不仅方便使用，同时也便于阅读。

Markdown

- 标题

Markdown支持两种标题语法：一种是利用“=”（最高等级标题）和“-”（第二等级标题），例如：

这是 H1

===

这是 H2

任何数量的“=”和“-”都是等效的，显示效果如下：

这是 H1

这是 H2

Markdown

- 标题

另一种是利用在行首插入1至6个“#”号，对应1至6级标题，例如：

```
# 这是 H1
## 这是 H2 ##
##### 这是 H6 ###
```

为了美观，可以在标题的后面也加上“#”号，行尾的“#”号的数量可以不和行首的保持一致，行首的“#”号决定了标题的等级，显示效果如下：

这是 H1

这是 H2

这是 H6

Markdown

- 区块引用

Markdown可以使用类似email中的“>”的引用模式。在每一行前面加上“>”，或只在段落的第一行前面加上“>”都可以实现区块引用的效果，例如：

```
> 这是一个包含两个段落的区块引用。  
> 我还是第一段。  
>  
> 我是第二段。
```

这是一个包含两个段落的区块引用。我还是第一段。

我是第二段。

区块应用可以嵌套，例如

```
> 第一级  
>  
> > 第二级  
>  
> 回到第一级
```

第一级

第二级

回到第一级

Markdown

- 列表

对于无序的列表，可以使用星号“*”，加号“+”或减号“-”作为标记，不同符号的效果是相同的；对于有序的列表，可以使用数字紧跟一个英文句号作为标记，数字的连续型和正确性并不影响有序列表的输出效果，但Markdown会将第一个数字作为起始值。

- * 红色

- * 绿色

- * 蓝色

- + Leo

- + Jack

1. 红色

2. 绿色

4. 蓝色

- 红色

- 绿色

- 蓝色

- Leo

- Jack

1. 红色

2. 绿色

3. 蓝色

Markdown

- 代码

对于和程序相关的写作通常会包含大量的代码块，通常对于这些代码我们不希望以一般的格式对其进行排版，而是希望代码块保持原始的样子。

在Markdown中插入代码块，需要对代码块缩进4个空格或1个制表符：

```
# Here is an example
Cubic <- function(num) {
  num * num * num
}
```

```
```{r}
Here is an example
Cubic <- function(num) {
 num * num * num
}
```
```

Markdown中使用单引号将文本包围起来
可以将行内文本转换成代码：

使用`print()`函数打印信息。

Markdown会将其转换为：

使用print()函数打印信息。

Markdown

- 分隔线

在Markdown中可以使用3个及以上的星号“*”，减号“-”或下划线“_”来建立一个分隔线。构建分割线的行内不能够有其他的符号，但在星号，减号和下划线之间可以插入空格，例如：

```
* * *  
***  
*****  
- - -  
-----
```

分割线的显示效果如下：

Markdown

- 链接

Markdown支持两种样式的链接语法：行内样式和参考样式。对于任何一种样式，链接文字都需要用方括号“[]”进行标记。

对于行内样式的链接，仅需在方括号的标记后面紧跟一对圆括号，并在圆括号内插入网址链接即可。如果需要对连接加上标题文字，可以在链接后面有双引号将标题问题包起来即可。例如：

```
[Google](https://www.google.com "Google主页")是一家美国的跨国科技企业。
```

Markdown会将其转换为：

Google是一家美国的跨国科技企业。

Markdown

- 强调

Markdown中使用星号“*”和下划线“_”作为强调的标记符号。被单个星号或下划线包围的文字转换后会被标签包围，被两个星号或下划线包围的文字转换后会被包围，例如：

```
*single asterisks*  
_单个下划线_  
**两个星号**  
__double underscores__
```

Markdown会将其转换为：

single asterisks
单个下划线
两个星号
double underscores

强调标记可以插在文档的任意位置，但需要注意以什么符号开始就需要以什么符号结束。同时需要保证在星号和下划线的两边不能出现空白，否则他们会被当做普通符号处理。如果需要插入一对星号或下划线的时候，可以在其前面添加反斜杠“\”进行转义。

Markdown

- 图片

Markdown中插入图片的方式和插入链接的方式类似，图片插入语法示例如下：

```
![替代文本](/path/to/img.jpg)  
![替代文本](/path/to/img.png "可选标题")
```

其中方括号内为图片的替换文本，小括号内为图片的路径，后面可以选择性的加上标题。

目前Markdown暂时还无法制定图片的宽度和高度，如果需要的话，可以使用原生的HTML标签。

Markdown

- 表格

表格并不是Markdown核心规范的一部分，但是GFM（Github Flavored Markdown）一部分。在Markdown中，表格的示例如下：

| Item | Quantity | Price |
|--------|----------|--------|
| Water | 10 | \$0.99 |
| Cheese | 2 | \$2.99 |
| Egg | 6 | \$0.49 |

| Item | Quantity | Price |
|--------|----------|--------|
| Water | 10 | \$0.99 |
| Cheese | 2 | \$2.99 |
| Egg | 6 | \$0.49 |

其中列之间用“|”进行分割，表头和数据之间用单独的一行进行分割，并且每列内“-”的个数至少为3个。表头和数据行的分割行内，可以使用英文冒号“:”定义数据列的对齐方式，默认（不添加“:”）为左对齐，在“-”两个均添加一个“:”为居中对齐，仅在“-”的右侧添加一个“:”为右对齐。

Markdown

- 转义字符

在Markdown中有很多符号最为了Markdown语法的标记符号，因此当需要将这些符号作为普通文本是，需要在其前面加上反斜杠进行转义。需要在其前面加上反斜杠进行转义的符号有：

- \ 反斜线
- ` 反引号
- * 星号
- _ 下划线
- {} 花括号
- [] 方括号
- () 小括号
- # 井字号
- + 加号
- - 减号
- . 英文句号
- ! 惊叹号

R Markdown

R Markdown

R Markdown是一种利用R语言创建动态文档，演示文稿和报告的语言。它包含了Markdown的核心语法和嵌入式的R语言代码块，从而可以使R代码的结果直接包含在最终的文档中。R Markdown文档可以完全被重现，因为任何时刻底层的R语言代码或数据发生改变，最终生成的文档也被重新生成。R Markdown包含了很多增强特性：

1. 多种输出格式（包括HTML，PDF和MS WORD）。
2. 支持创作Beamer，ioslides和Slidy样式的演示文档。
3. 支持表格和参考文献的Markdown语法。
4. 用于个性化HTML和PDF的接口（包括CSS，headers和footers）。
5. 用于个性化PDF输出的原始LaTeX支持。
6. 利用R脚本编译HTML，PDF或MS WORD笔记。
7. 可扩展性：创建自定义模板或定义新的输出格式。
8. 利用Shiny创建可以交互性R Markdown文档。

尽管用户可以按照R Markdown的语法要求书写R Markdown文档，并利用rmakrdwn扩展包中的render()函数渲染文档并生成最终的文档，但利用knitr扩展包会让生成一份优美的文档变得更加“优美”。

R Markdown

- R代码块

在R Markdown中，通过如下方式嵌入R代码块：

```
```{r}
summary(cars)
```
```

```
summary(cars)
```

| ## | speed | dist |
|----|--------------|----------------|
| ## | Min. : 4.0 | Min. : 2.00 |
| ## | 1st Qu.:12.0 | 1st Qu.: 26.00 |
| ## | Median :15.0 | Median : 36.00 |
| ## | Mean :15.4 | Mean : 42.98 |
| ## | 3rd Qu.:19.0 | 3rd Qu.: 56.00 |
| ## | Max. :25.0 | Max. :120.00 |

在R代码块中可以制定相应参数，例如：

```
```{r, echo=FALSE, fig.width=8, fig.height=6}
plot(cars)
```
```

R Markdown

- 行内执行R代码

R Markdown还支持在文本行内执行R代码，待执行的R代码需要用反引号包围起来并在开头添加“r”，例如：

```
行内计算的结果为`r 1+1`
```

则输出的结果为：

行内计算的结果为2

R Markdown

- LaTeX和MathML公式

在R Markdown中可以嵌入LaTeX或MathML公式，例如：

1. `$equation$`用于行内公式（注意\$符号内侧不能由空白符号）。

例如： `$a = \sum_{i=1}^n \{i\}$`显示为： $a = \sum_{i=1}^n i$

2. `$$equation$$`用于显示的公式。

例如： `$$a = \sum_{i=1}^n \{i\}$$`显示为：

$$a = \sum_{i=1}^n i$$

3. `$$`用于MathML公式。

R Markdown

- 图像编码和打包

当R Markdown被转化为HTML时，文件中用相应图像将直接被打包进最终生成的HTML文件中。图像通过base64编码，然后利用data URI插入被编码的图像，从而产生一个独立的web页面。图像编码和打包使得在分享时不需要再打包任何其他附件文件，可以将生成的HTML文件作为邮件附件，在公共文件夹中分享，或者在web服务中将其配置成单独的HTML文件。

[1] R Markdown Reference Guide: https://www.rstudio.org/links/r_markdown_reference_guide

[2] R Markdown Cheatsheet: https://www.rstudio.org/links/r_markdown_cheat_sheet

输出类型

- 报告
 - 自带：多种输出格式：HTML, PDF (需TeX环境支持，例如：TeX Live), Word
 - **rticles**：多种 LaTeX 排版格式文档，其中 CTeX Documents 支持中文
 - **tufte**：Tufte格式排版 (3 种输出类型，中文支持)，**示例**
- 幻灯片
 - 自带：HTML (ioslide, Slidy), PDF (Beamer)
 - **xaringan**
- 博客
 - **blogdown**
- 书籍
 - **bookdown**

[1] R Markdown Formats: <https://rmarkdown.rstudio.com/formats.html>

[2] R Markdown Gallery: <https://rmarkdown.rstudio.com/gallery.html>

R Notebook

R Notebook

- [R Notebook]: https://rmarkdown.rstudio.com/r_notebooks.html
- [Jupyter with R]: 参见配置手册

写在后面

谢益辉 对可重复研究的相关总结：

- Sweave：打造一个可重复的统计研究流程
- Sweave 后传：统计报告中的大规模计算与缓存
- 真理在缩水，还是上帝在掷骰子？
- knitr 与可重复的统计研究（花絮篇）

Thanks



本作品采用 **CC BY-NC-SA 4.0** 进行许可