

```
language=R, backgroundcolor=backcolour, numbers=left, numberstyle= , commentstyle= red!50!green!50!blue!50, rulesepcolor= red!20!green!20!blue!20 , escapeinside=" , xleftmargin=2em,xrightmargin=2em, aboveskip=1em, framexleftmargin=2em
```

§0.1 R Markdown简介

R Markdown是在R语言中生成数据分析报告的利器。它提供了数据分析报告的模版，在模版中我们可以很方便地插入代码、运行结果和解释性的文字等，经过knitr包和pandoc处理后，可以生成各种格式的分析报告，如html和pdf等。Markdown是一种轻量级的标记语言，能够用简洁的代码，编写出一份美观的文档。R Markdown的语法基本上承袭了Markdown，相比Latex和html更容易上手，而且编写文档的同时，R Markdown可以插入多种编程语言尤其是r语言的代码块，在knitr生成报告时运行代码并展示代码运行结果，这使得我们能够轻松地生成一份包含着代码、运行结果和解释性文字的精美报告。除了一个固定的数据分析报告模版外，R Markdown还有很多扩展的模版，如prettydoc、rticles、xaringan等包，分别提供了很多html报告、pdf文档和presentation的模版。

§0.2 R Markdown安装

R Markdown 配合RStudio 使用很方便，默认大家已经安装配置好了R 语言和RStudio，在使用R Markdown生成报告前，我们需要先安装R Markdown 包。运行以下代码安装R Markdown 包：

```
install.packages("rmarkdown")
```

安装完R Markdown 包后，已经能够生成基本的html报告了，但是想要生成pdf报告，还需要Latex的支持，在电脑上配置Latex。由于Latex 对中文的支持不如英文，需要在R中生成中文pdf文档较为麻烦。手动在电脑上配置Latex 后，不论配置了CTeX、Tex Live还是MikTeX，如果要通过R Markdown生成中文pdf还需要手动安装一些缺失的宏包，也经常容易遇到一些难以处理的报错，耗费太多的时间。谢益辉编写的TinyTex包正是为此而生的，相比Latex 发行版，TinyTex更加轻巧，在R Markdown 生成中文pdf时，TinyTex不仅提供Latex 上的支持，还能自动安装Latex缺失的宏包，极大地节省了时间。因此推荐安装TinyTex包，可以很轻松地生成一份中文pdf 报告。运行以下代码安装TinyTex 包(需要较长的时间)：

```
install.packages("tinytex") tinytex::install_tinytex()
```

§0.3 编写

§0.3.1 文档组成

R Markdown 文档的文件扩展名为.Rmd。在RStudio中，通过File -> New File -> R Markdown 创建一个默认的R Markdown 文档，即.Rmd 文件。R Markdown 文档有三个重要的组成部分——元数据、文字、代码。元数据在文档的最上方，采用的语法是YAML，以“---”开头和结尾，可以编辑文章的标题、作者、日期和输出格式等。R Markdown 文档的文字大体有段落、标题、列表等，使用的语法是Markdown。R Markdown中的代码一般分块存储在代码框中，代码框可以设置是否执行代码、是否显示代码、是否显示代码运行结果等。在学习R Markdown 文档的结构和语法之前，不妨先新建一个默认的R Markdown文档，点击页面右上方的knit 按钮，即可生成一份html 报告，对照着R Markdown文档和生成html报告学习下面的内容将更为容易。下面我们将分别介绍R Markdown 中的元数据、文字和代码。

§0.3.2 元数据

R Markdown的元数据位于文档的正上方，是对最终输出文档的一个基本设置，主要包括title、author、date和output，分别代表文章的标题、作者、日期和输出格式。一个元数据的基本框架如下：

```
--- title: author: date: output: ---
```

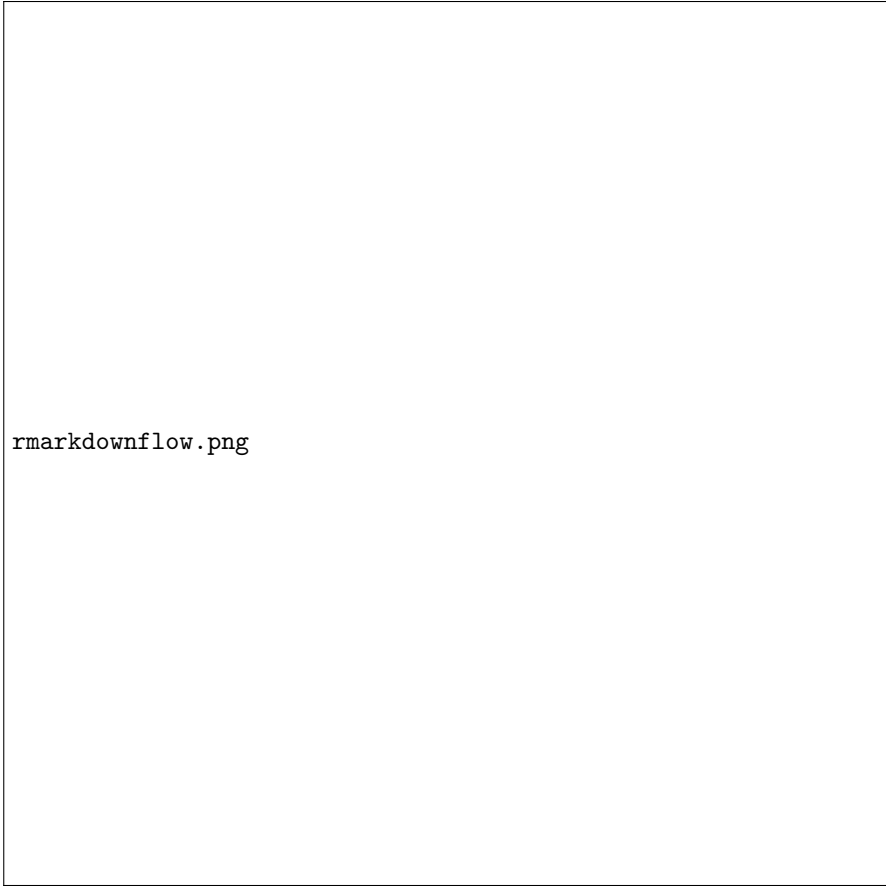
title、author和date比较容易理解，output则较为复杂，涉及最终输出文档的格式，常用的格式有html和pdf。在介绍如何生成html 报告和pdf 文档之前，我们先了解一下R Markdown生成报告的原理。

图??展示了R Markdown生成一份报告的步骤。以一份编写好的R Markdown 文档为例，点击knit按钮之后，R Markdown 文档首先经knitr 包处理，生成Markdown 文档，接着借助pandoc将Markdown文档转化为各种格式的报告，其中最常用的为html 和pdf格式。而元数据中的output就是对最终生成报告格式的设置。下面将分别介绍如何html报告和pdf 文档的元数据如何设置。

§0.3.2.1 html报告——html_document

元数据中的output可以规定生成的报告类型，想要生成html格式的报告，需要将output设置为html_document。在html_document 下，又有html 主题、是否生成目录等设置。下面是一个生成html_document 的R Markdown 文档的元数据：

```
--- title: "Your Document Title" author: "Document Author" date: "r Sys.Date()"
output: html_document : theme : default toc : yes toc_depth : 3 ---
```



rmarkdownflow.png

`theme`表示html_document使用的主题, `toc`表示html_document中是否显示目录, `toc_depth`表示目录的深度, 即目录涵盖多少级的标题。在R 中运行以下代码可以查看html_document 下的参数:

```
?html_document
```

§0.3.2.2 html报告——html_pretty

prettydoc包是R Markdown的一个扩展, 包含了数个精美的html报告模版, 安装prettydoc 包后, 将output设置为html_pretty, 调用prettydoc 包的模版。运行以下代码, 安装prettydoc包:

```
install.packages("prettydoc")
```

安装好prettydoc包之后, 可以生成html_pretty报告。一个生成html_pretty 的R Markdown文档的元数据如下:

```
— title: "Your Document Title" author: "Document Author" date: "r Sys.Date()"
output: html_pretty : theme : caymantoc : yestoc_depth : 3 ---
```

要生成一份html_pretty报告,可以选择修改R Markdown文档的元数据的output;也可以在新建R Markdown 文档时,按照File -> New File -> R Markdown -> From Template -> Lightweight and Pretty Document 的点击顺序,建立一个默认html_pretty文档;或者点击knit 按钮之后,在下拉选项中选择Knit to html_pretty。

§0.3.2.3 pdf文档——pdf_document

R Markdown生成pdf_document需要Latex的支持,默认大家安装好了TinyTex包,首先将元数据的output设置为pdf_document,在Latex环境下,能够生成一份不带中文的pdf_document,由于Latex对中文的支持不如英文,因此想要生成中文pdf还需要对元数据进行进一步的设置,生成中文pdf_document的元数据如下:

```
— title: "Your Document Title" author: "Document Author" date: "r Sys.Date()"
output: pdf_document : latex_engine : xelatexCJKmainfont : MicrosoftYaHei ---
```

latex_engine是对生成pdf_document时使用的Latex的命令的设置,默认的设置是pdflatex,但是xelatex 在对中文的支持上做的更好,因此选择xelatex。CJKmainfont是对pdf_document的中文字体的设置,如果不设置字体,也无法显示中文。更多关于pdf_document的参数设置,运行以下代码:

```
?pdf_document
```

§0.3.3 文字

R Markdown中的文字使用Markdown的语法,下面介绍Markdown中一些基本的语法,R Markdown继承了这些语法,可以使用这些语法对最终输出文档中的文字进行排版。

§0.3.3.1 段落

R Markdown中的段落之间至少要相隔一空行,由于最终输出的html报告和pdf文档的段首都不会有缩进,因此也可以在段尾打两个空格再换行,不需要空出一整个空行,但效果和另起一个段落是一致的。

§0.3.3.2 行内文字格式

倾斜:

```
*text*
```

加粗:

text

上标: t为上标

text^t

下标: t为下标

text_t

行内显示代码: 会赋予code以特殊的格式, 但是不运行代码, 只呈现代码

`code`

脚注: footnote为脚注内容

text^[footnote]

§0.3.3.3 插入网页链接和图片

网页链接: text为最终文档显示的文字, 点击会链接到url

[text](url)

图片: 相比网页链接只多了一个“!”

![text](url)

§0.3.3.4 分级标题

标题以#开头, #的数量规定了标题等级

一级标题 二级标题 三级标题 四级标题 五级标题等等

§0.3.3.5 列表

列表分为无序列表和有序列表, 与正文之间的间隔至少要有一个空行。

无序列表: 以*或者+或者-开头

这是无序列表的示例:

* 这是无序列表的第一点* 这是无序列表的第二点* 这是无序列表的第三点

有序列表: 以number.开头, 例如1. text

这是有序列表的示例:

1. 这是有序列表的第一点2. 这是有序列表的第二点3. 这是有序列表的第三点

§0.3.3.6 引用

以<开头, 例,

<”这里写引用的内容” < <”这里写作者”

§0.3.3.7 数学公式

R Markdown 中的数学公式大体有三种形式：

1. 行内公式，用一对\$括起来，公式的语法为Latex语法。
2. 单独成行公式，用一对\$\$括起来，公式的语法为Latex语法。
3. 建立一个公式环境，用\$\$

begin{}开头，

end{}\$\$结尾，公式的语法为Latex 语法。

§0.3.4 代码

§0.3.4.1 代码框

新建一个代码框，可以点击R Markdown 编辑界面右上方的insert按钮，选择插入R 语言代码框，或者使用快捷键Ctrl + Alt + I(Mac 用户快捷键为Cmd + Option + I)，或者在文档中代码框键入三个`啁蜚r`，作为代码框开头，再键入三个`啁蜚

“r code “

§0.3.4.2 行内代码

行内代码以r 开头，以`啁蜚r`作为代码框开头，再键入三个`啁蜚

a的值为‘r a’

§0.3.4.3 代码框设置

在R Markdown中，可以对代码框进行设置，如是否运行代码、是否显示代码、是否显示代码结果、代码生成的图片的宽高等等。以是否显示代码为例，{r echo = FALSE}代表不显示代码，即在最终文档中不会显示该代码框，但是代码仍然会运行。代码框的写法如下：

“r echo = FALSE a j- 1;a “

下面列举一些常用的代码框选项：

eval: 是否运行代码

echo: 是否显示代码

message: 是否显示message

warning: 是否显示warning

error: 是否显示error

fig.weight、fig.height: 代码生成的图片在最终文档中宽度和高度

谢益辉编写的knitr包提供了大量代码框选项的支持，更多的代码框选项详见网址<https://yihui.name/knitr/options>。

如果想对整个文档的所有代码框进行相同的设置，可以在文档的元数据下面，建立一个全局设置的代码框，注意全局设置的优先级低于代码框的个性设置，例如，可以设置文档中全部的代码框都不显示运行结果，在特定的代码框处，设置显示运行结果，则最终输出的文档中只有特定的代码框显示运行结果。全局代码框的格式为：

```
“r setup, include=FALSE, ... knitr::opts_chunk$set(echo = TRUE,...) “
```

§0.3.4.4 表格

在R Markdown 中，knitr包提供了一个插入表格的函数——kable()，使用kable()函数可以轻松地插入一个美观的表格，代码如下：

```
“r iris[1:4, ] knitr::kable(iris[1:4, ], caption = ”美观的表格”) “
```

除knitr包外，stargazer包也提供了插入正规表格的函数——stargazer。stargazer包是Marek Hlavac编写的，用于将R 中的数据框转化为Latex 或者html语法的表格，即向stargazer包输入数据框，能得到该数据框按照某一模版的Latex或者html代码，将该代码在对应的语言环境下运行，可以得到数据框的一个较为正规的表格。代码如下：

```
stargazer::stargazer(iris[1:4, ], summary = FALSE, title = ”XXX”, ...) s-
targazer::stargazer(iris[1:4, ], summary = FALSE, type = ”html”)
```

stargazer函数的第一个参数为输入的数据框，summary = FALSE表示输出数据框本身而不是数据框的summary，title是输出表格的标题，type 默认取值为”default”，表示输出Latex 代码，取值”html”可以输出数据框的html 代码。上述代码的输出结果是相应的代码，代码用Latex和html编译后才能输出表格。因此，在使用Latex 编写pdf 时，我们可以选择将函数的输出结果复制到Latex环境中，将R中的数据框，插入我们的Latex 文档中。如果编写的是R Markdown文档，只需要修改代码框的设置，即可在最终结果中直接输出最终的表格而不是Latex 或html 代码。以输出pdf文档的R Markdown文档为例，代码如下：

```
“r results=”asis” stargazer::stargazer(iris[1:4, ], summary = FALSE) “
```

代码框设置results = ”asis”表示将stargazer函数生成的Latex代码，直接编译成表格，如果不对代码框的results 加以设置，在最终pdf文档中将输出stargazer函数的原输出结果，即数据框的Latex代码。

除将数据框输出为表格外，stargazer包还能将模型结果输出为表格。stargazer包对R 中各种包的各种建模函数提供了广泛的支持，可以将各种函数的建模结果输出为表格，以普通线性回归建模函数——lm()为例，代码如下：

```
fit1 <- lm(rating ~ complaints + privileges + learning, data=attitude) fit2 <-
lm(rating ~ complaints + privileges + learning + raises + critical, data = attitude)
stargazer(fit1, fit2, align = TRUE)
```

同样地，上面的代码输出的为Latex语法的表格，如果想要在R Markdown 中直接呈现最终输出的表格，需要对代码框的results 进行设置。有关stargazer函数的更多参数，可以运行以下代码，查看help文档：

```
?stargazer::stargazer
```

§0.4 输出

在编写完R Markdown文档后，点击界面右上角的knit按钮，R Markdown 就会自动生成一份精美的报告了。如果想要进一步学习R Markdown，可以参考以下网址或者文档：

官网：<https://rmarkdown.rstudio.com/>

bookdown: <https://bookdown.org/yihui/rmarkdown/>

knitr: <https://yihui.name/knitr/>

cheatsheet: <https://www.rstudio.com/wp-content/uploads/2016/03/rmarkdown-cheatsheet-2.0.pdf>

reference: <https://www.rstudio.com/wp-content/uploads/2015/03/rmarkdown-reference.pdf>