

如何用 R Markdown 写学术文档？

Shujia Wong

摘要： 这算是用 R Markdown 学习 R 和写作的学习笔记吧。主要包括：1. 用 Rmarkdown 写科研笔记或一般学术文档；2. 用 bookdown 写书或论文；3. 记录 Rmarkdown 的一些用法。

关键词： Rmarkdown; bookdown; bookdownplus

1 介绍

1.1 什么是 R Markdown？

R Markdown (Allaire et al., 2018) 是结合 Markdown 和 R(R Core Team, 2018) 语言的写作软件。

Markdown 是轻量级、纯文本、超简单的书写格式。

R Markdown 可以做什么？

- 语法简单。作者基本上无需关心排版问题，只要专心写作就可以了。
- 计算结果动态生成。作者不必手动拷贝粘贴代码结果或者生成的表格、图片等。
- 易于修改。写作过程中如需修改某处，全文相应变动会自动生成，包括软件运行的结果（图形等）。
- 比 Word 更美观，比 LaTeX 更易用。
- 方便地插入目录、图表、脚注等。Bookdown 扩展功能可以交叉引用、索引。
- 方便地插入数学公式、参考文献、R 代码。
- 可以生成漂亮的 pdf、word、epub、网页和幻灯片等多种文件格式。
- 写作及结果具有可重复性。
- 支持多语言，包括 R, C/C++, Python, Fortran, Julia, Shell scripts 和 SQL 等。
- R 和 Markdown 都是开源免费的。

Rmarkdown 使得数据分析代码可以与文档混编，具体语法请参看《R Markdown 语法简介》。

下面是回归模型的简单例子。在 R 中，回归模型可以用非常简单的一行代码搞定：`lm(y~x, data)`,

```
options(digits = 4)
fit = lm(dist ~ speed, data = cars)
coef(summary(fit))
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-17.579	6.7584	-2.601	1.232e-02
speed	3.932	0.4155	9.464	1.490e-12

```
b = coef(fit)
```

上面回归方程中的斜率是 3.9324，完整的回归方程为：

$$Y = -17.5791 + 3.9324x$$

散点图和回归直线 (见图1):

```
par(mar = c(4, 4, .1, .1), las = 1)
plot(cars, pch = 19)
abline(fit, col = 'red')
```

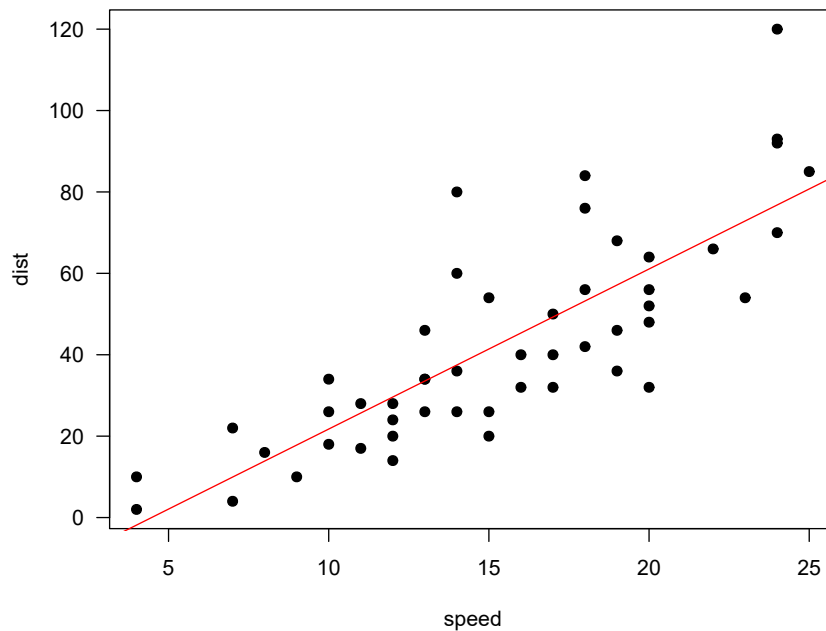


图 1: cars 数据散点图以及回归直线。

1.2 系统环境

需要安装的软件:

- R: <https://www.r-project.org/>
- RStudio: <https://www.rstudio.com/>
- Bookdown: <https://bookdown.org/>

安装 Bookdown (Xie, 2018a) 及其相依软件包基本上就 OK 了, 包括了 rmarkdown (Allaire et al., 2018)、knitr (Xie, 2018b)、rticles 等, 如果有缺失的, 运行时会提示安装。

可以安装稳定版: `install.packages('bookdown')`

或开发版: `devtools::install_github('rstudio/bookdown')`

1.3 RStudio 基本设置

打开 Tools→global options, 然后

- 点击 `sweave`, 在 `weave rnw files using` 选择 **knitr**
- 在 `Typeset LaTeX into PDF using` 选择 **XeLaTeX**
- 在 `Code -> Saving -> Default text Encoding` 选择 UTF-8

2 用 R Markdown 写简单学术文档

所谓“简单”学术文档, 是指满足了学术文档的基本要求, 包括标题、公式、图表、参考文献以及自动编号等, 但是不能交互引用。这种情况下 RMarkdown 具有独特优势, 可以直接上手, 输出结

果丰富。

1. 新建一个 R Markdown 文档。在打开界面可输入题目和作者姓名，输出格式可选择 `html`，`pdf` 或 `word`。
2. 简单设置。点击 `Knit` 旁边的齿轮按钮，在 `Output Options` 可做更多选择，比如可勾选 `Include table of contents`，表示建立目录。勾选 `Number section headings` 表示章节题目包含顺序数字。在 `Figures` 勾选 `Render figure captions` 表示图形会显示标题并自动排序。
3. 结果输出。
 - 如果文档是纯英文的，可以通过 `knit` 转为 `html`，`pdf` 或 `word` 任何格式都没问题。不过，如果你用中文 Windows，则日期最好用 `"` r Sys.Date() `"` 代替或直接输入英文日期，否则可能会乱码。
 - 如果文档是中文的，则
 - 转 `html` 没有任何问题。
 - 转 `word`，一定要在开始时指定，或在文档的 `output:` 下面指明 `word_document:`，否则标题和日期会是乱码。
 - 中文文档不能直接转 `pdf`。

中文文档如何转 pdf?

要转为 pdf 文档，必须安装软件包 `rticles`。用法很简单：打开 Rmarkdown 时，选择 `From Template`，从模版中选择 `CTex Documents` 即可。

一般文档开头（称为 `yaml`）是这样的：

```
---
title: " 你的论文题目"
author: " 你的姓名"
date: "2018-09-17"
output:
  html_document:
    fig_caption: yes          # 文档格式为 html
    number_sections: yes      # 包括图形标题
    toc: yes                  # 章节数字顺序
    toc_depth: 3              # 显示目录
    toc_float: True           # 三级目录
                                # 目录作为侧边栏
bibliography: reference.bib  # 参考文献文件名
bibli0-style: apalike        # 参考文献格式
link-citations: yes          # 参考文献链接
colorlinks: yes              # 链接颜色
#lot: yes                    # 表格列表
#lof: yes                    # 图形列表
---
```

3 用 R Markdown 写学术文档

前面介绍的 R Markdown 文档是没有交叉引用功能的，即公式、图形、表格或章节之间在正文的引用不能自动编号。

为了实现交叉引用功能，可以扩展到 `bookdown`。

撰写可以交叉引用的网页文档：

把 `html_document:` 换成 `bookdown::html_document2:` 即可，其它不变。

撰写可以交叉引用的 pdf 文档：

如果文档是纯英文的，把 `html_document:` 换成 `bookdown::pdf_document2:` 即可，其它不变。如果文档含有中文字符，还是老老实实用软件包 `bookdown` 或 `bookdownplus` 吧。

撰写可以交叉引用的 **docx** 文档:

一样把 `word_document:` 换成 `bookdown::word_document2:`。但是直接产生的 word 文档格式不一定满足要求, 如标题是蓝色的等等。

可以制作一个**参考文档**: 由 R Markdown 产生一个 word 文档, 然后对该文档的**样式**进行编辑(编辑其它功能无效)。

我已经编辑了一个参考文档: `word_style.docx`, 你可以结合需要自行再通过样式修改。

我的 `yaml` 是这样的:

```
---
title: "Word 参考文档"
author:
  - WSJ
  - College of Economics, Shenzhen University
date: "2018-09-17"
output:
  bookdown::word_document2:
    reference_docx: template/word_style.docx
    fig_caption: yes
bibliography: [bib/packages.bib,bib/book.bib]
csl: csl/chinese-gb7714-1987-numeric.csl
#bibli0-style: apalike
link-citations: yes
colorlinks: yes
#lot: yes
#lof: yes
---
```

如何修改参考文献格式?

注释掉 `#bibli0-style: apalike`, 然后增加一行:

`csl: csl/chinese-gb7714-1987-numeric.csl`

4 用 Bookdown 写书或长文

长篇论文(书籍、毕业论文)需要分章节以及交叉引用, 结构复杂, 简单的一个 Rmarkdown 文档可能不能胜任。

一本 **bookdown** 书含有多个章节, 每个章节写在单独的 `.Rmd` 文件, 起始部分为该章节标题 `# Chapter Title`。

如果含有中文, 所有 R Markdown 文档都必须用 UTF-8 编码保存。

以下是一本书的典型例子:

- `index.Rmd`

```
# Preface {-}
```

```
In this book, we will introduce an interesting
method.
```

- `01-intro.Rmd`

```
# Introduction
```

```
This chapter is an overview of the methods that
we propose to solve an important problem.
```

- 02-literature.Rmd

```
# Literature

Here is a review of existing methods.
```

- 03-method.Rmd

```
# Methods

We describe our methods in this chapter.
```

- 04-application.Rmd

```
# Applications

Some _significant_ applications are demonstrated
in this chapter.

## Example one

## Example two
```

- 05-summary.Rmd

```
# Final Words

We have finished a nice book.
```

具体用法:

- 下载模板: 在模板下载网页 (github) 的右上角, 点击 **Clone or download** 下载压缩文件, 解压到工作目录。
 - 英文模版 <https://github.com/rstudio/bookdown-demo>。
 - 中文模版 <https://github.com/yihui/bookdown-chinese>
- 用 RStudio 打开文件 `bookdown-demo.Rproj` 或 `bookdown-chinese.Rproj`, 然后在右上角点击 **Build**, 下一行 **Build Book**, 然后选择相应格式 (pdf,epub,word,gitbook) 即可得到模板文件。
- 根据自己需要修改相关文件, 保存。运行 **Build Book** 即可得到你自己的书籍。

其中中文书可输出格式 `gitbook` 网页、`pdf`、`word` 和 `epub`。

英文的 `demo` 里默认没有 `word` 格式的输出, 要自行在 `_output.yml` 里添加一行:

```
bookdown::word_document2: default
```

更多说明请参看 Xie Yihui (2018): *bookdown: Authoring Books and Technical Documents with R Markdown* (<https://bookdown.org/yihui/bookdown/>)

5 用 Bookdownplus 模版快速上手

Bookdownplus(Zhao, 2017) 对 Bookdown 进行了重新配置, 制作了多种文本的模版, 包括中、英文论文, 北大、浙大等高校毕业论文等, 大大方便了各种类型的写作应用。

5.1 安装调用

软件包 `bookdownplus` 可参看 CRAN <https://cran.r-project.org/web/packages/bookdownplus/index.html>。也可以到 Github <https://github.com/pzhaonet/bookdownplus> 去下载相关文件。

```
安装稳定版:install.packages('bookdownplus')
```

或开发版: `devtools::install_github('pzhaonet/bookdownplus')`

调用软件包:

```
require(bookdownplus)
```

查看所有可用模版:

```
template()
[1] "article"          "article_mdpi"      "article_zh"
[4] "calendar"         "cchess"            "chemistry"
[7] "chemistry_zh"     "discussion"        "dnd_dev"
[10] "docsens"          "guitar"            "igo"
[13] "journal"          "mail"              "musix"
[16] "nonpar"           "nte_zh"            "poem"
[19] "poster"           "rbasics"           "skak"
[22] "thesis_classic"   "thesis_mypku_zh"   "thesis_pku_zh"
[25] "thesis_ubt"       "thesis_zju_zh"     "yihui_crc"
[28] "yihui_demo"       "yihui_mini"        "yihui_zh"
```

其中 `article` 是英文版学术论文模版, `article_zh` 是中文学术论文模版, `thesis_pku_zh` 是北京大学毕业论文模版, `thesis_zju_zh` 是浙江大学的。

6 R Markdown 简介

6.1 基本语法

6.1.1 行内语法

- 斜体: `text`, `_text_` or `*text*`
- 黑体: `text`, `__text__` or `**text**`
- 下标: 两个 `~` 包夹, `H~2~SO~4~` renders H_2SO_4
- 上标: 两个 `^` 包夹, `R^2^` renders R^2
- 显示代码: `code` 和 `code` 显示代码块
- 行内运行 R 代码: `10`, `100` 的开根号等于 `10`
- 链接网站: `[text](link)`, 例如 `RStudio`
- 插入图片: `![image title](path/to/image)`
- 脚注: `^[]`, e.g., 这里是脚注¹

6.1.2 模块语法

Section headers can be written after a number of pound signs, e.g.,

```
# First-level header
## Second-level header
### Third-level header
```

If you do not want a certain heading to be numbered, you can add `{-}` after the heading, e.g.,

```
# Preface {-}
```

Unordered list items start with `*`, `-`, or `+`, and you can nest one list within another list by indenting the sub-list by four spaces, e.g.,

```
- one item
- one item
- one item
```

¹This is a footnote.

- one item
- one item

The output is:

- one item
- one item
- one item
 - one item
 - one item

Ordered list items start with numbers (the rule for nested lists is the same as above), e.g.,

1. the first item
2. the second item
3. the third item

The output does not look too much different with the Markdown source:

1. the first item
2. the second item
3. the third item

Blockquotes are written after >, e.g.,

“I thoroughly disapprove of duels. If a man should challenge me, I would take him kindly and forgivingly by the hand and lead him to a quiet place and kill him.”

— Mark Twain

Plain code blocks can be written after three or more backticks, and you can also indent the blocks by four spaces, e.g.,

```

```
This text is displayed verbatim / preformatted
```

```

Or indent by four spaces:

```

    This text is displayed verbatim / preformatted

```

6.2 数学公式

行内公式 $y=x^2$ 显示 $y = x^2$ ，整行公式（display style）用两双美元符号之间表示， $f(k) = \binom{n}{k} p^k (1-p)^{n-k}$ ，结果：

$$f(k) = \binom{n}{k} p^k (1-p)^{n-k}$$

You can also use math environments inside $$ or $$, e.g.,

```

 $\begin{array}{ccc}
x_{11} & x_{12} & x_{13} \\
x_{21} & x_{22} & x_{23}
\end{array}$ 

```

$$\begin{array}{ccc} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \end{array}$$

```


$$X = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & x_3 \end{bmatrix}$$


```

$$X = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & x_3 \end{bmatrix}$$

```


$$\Theta = \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix}$$


```

$$\Theta = \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix}$$

```


$$\begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc$$


```

$$\begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc$$

6.3 R 代码块

You can insert an R code chunk either using the RStudio toolbar (the Insert button) or the keyboard shortcut `Ctrl + Alt + I` (`Cmd + Option + I` on macOS).

There are a lot of things you can do in a code chunk: you can produce text output, tables, or graphics. You have fine control over all these output via chunk options, which can be provided inside the curly braces (between ```{r` and `}`). For example, you can choose hide text output via the chunk option `results = 'hide'`, or set the figure height to 4 inches via `fig.height = 4`. Chunk options are separated by commas, e.g.,

```
``{r, chunk-label, results='hide', fig.height=4}
```

The value of a chunk option can be an arbitrary R expression, which makes chunk options extremely flexible. For example, the chunk option `eval` controls whether to evaluate (execute) a code chunk, and you may conditionally evaluate a chunk via a variable defined previously, e.g.,

```

``{r}
# execute code if the date is later than a specified day
do_it = Sys.Date() > '2018-02-14'
...

``{r, eval=do_it}
x = rnorm(100)
...

```

There are a large number of chunk options in **knitr** documented at <https://yihui.name/knitr/options>. We list a subset of them below:

- `eval`: Whether to evaluate a code chunk.
- `echo`: Whether to echo the source code in the output document (someone may not prefer reading your smart source code but only results).

- **results:** When set to 'hide', text output will be hidden; when set to 'asis', text output is written “as-is”, e.g., you can write out raw Markdown text from R code (like `cat('**Markdown** is cool.\n')`). By default, text output will be wrapped in verbatim elements (typically plain code blocks).
- **collapse:** Whether to merge text output and source code into a single code block in the output. This is mostly cosmetic: `collapse = TRUE` makes the output more compact, since the R source code and its text output are displayed in a single output block. The default `collapse = FALSE` means R expressions and their text output are separated into different blocks.
- **warning, message, and error:** Whether to show warnings, messages, and errors in the output document. Note that if you set `error = FALSE`, `rmarkdown::render()` will halt on error in a code chunk, and the error will be displayed in the R console. Similarly, when `warning = FALSE` or `message = FALSE`, these messages will be shown in the R console.
- **include:** Whether to include anything from a code chunk in the output document. When `include = FALSE`, this whole code chunk is excluded in the output, but note that it will still be evaluated if `eval = TRUE`. When you are trying to set `echo = FALSE`, `results = 'hide'`, `warning = FALSE`, and `message = FALSE`, chances are you simply mean a single option `include = FALSE` instead of suppressing different types of text output individually.
- **cache:** Whether to enable caching. If caching is enabled, the same code chunk will not be evaluated the next time the document is compiled (if the code chunk was not modified), which can save you time. However, I want to honestly remind you of the two hard problems in computer science (via Phil Karlton): naming things, and cache invalidation. Caching can be handy but also tricky sometimes.
- **fig.width and fig.height:** The (graphical device) size of R plots in inches. R plots in code chunks are first recorded via a graphical device in **knitr**, and then written out to files. You can also specify the two options together in a single chunk option `fig.dim`, e.g., `fig.dim = c(6, 4)` means `fig.width = 6` and `fig.height = 4`.
- **out.width and out.height:** The output size of R plots in the output document. These options may scale images. You can use percentages, e.g., `out.width = '80%'` means 80% of the page width.
- **fig.align:** The alignment of plots. It can be 'left', 'center', or 'right'.
- **dev:** The graphical device to record R plots. Typically it is 'pdf' for LaTeX output, and 'png' for HTML output, but you can certainly use other devices, such as 'svg' or 'jpeg'.
- **fig.cap:** The figure caption.
- **child:** You can include a child document in the main document. This option takes a path to an external file.

6.4 图形

6.4.1 R 代码块做图

代码块选项 `fig.asp` 表示图形的高宽比。如果宽度为 6 英寸 (`fig.width = 6`) 而 `fig.asp = 0.7`, 则图形的高度为 `fig.width * fig.asp = 6 * 0.7 = 4.2` (图2)。

```
par(mar = c(4, 4, .1, .1))
plot(pressure, pch = 19, type = 'b')
```

`fig.width` 和 `fig.height` 给出的是图形的实际大小。我们可以用 `out.width` 和 `out.height` 给出图形的输出大小。例如, `out.width = '70%'` **knitr** 会自动处理为 `.7\linewidth`, 即文本宽度的 70%

An example of `out.width = 70%` (图3)。

```
par(mar = c(4, 4, .1, .1))
plot(cars, pch = 19)
```

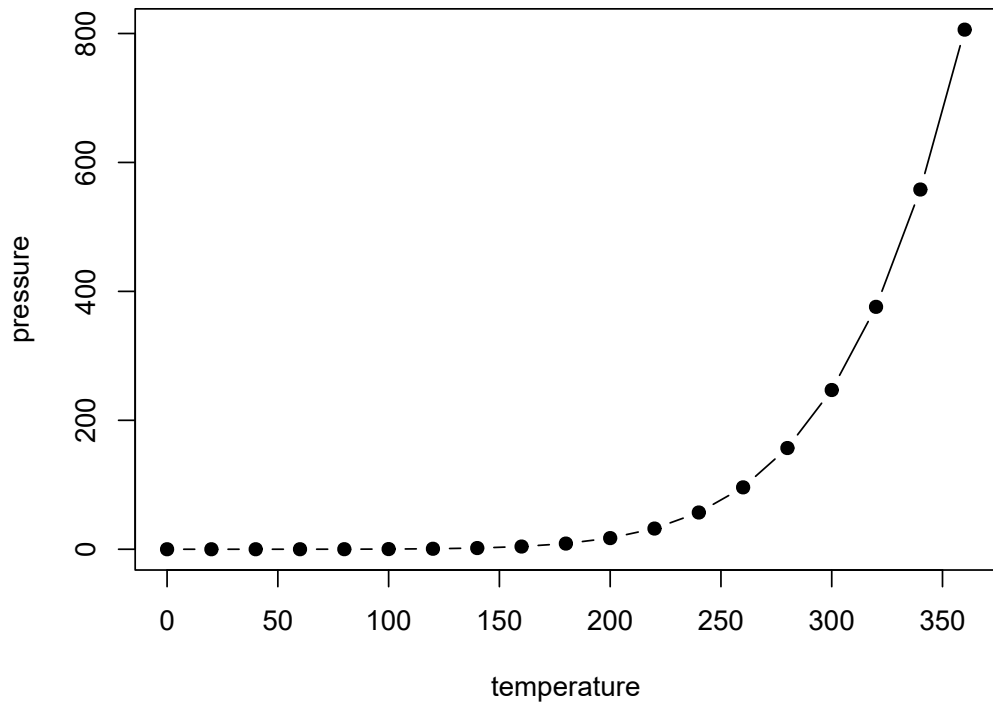


图 2: A figure example with the specified aspect ratio, width, and alignment.

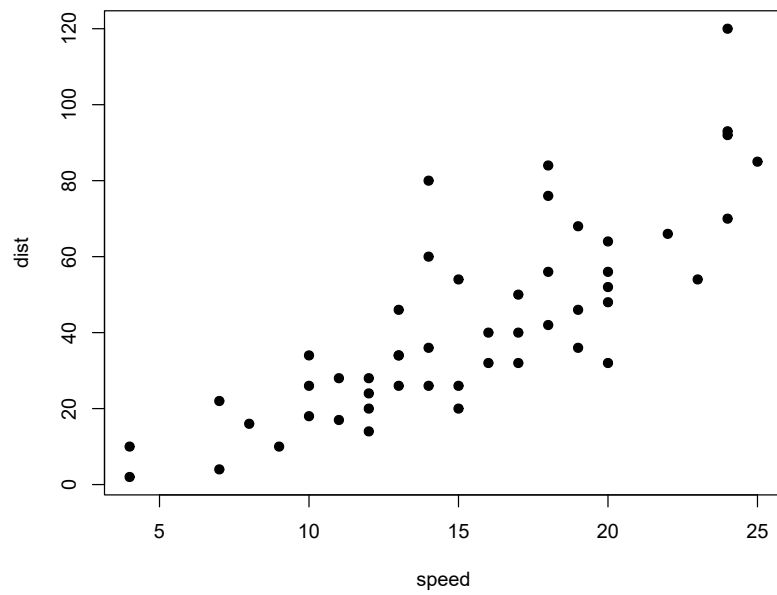


图 3: A figure example with a relative width 70%.

如果一幅图里有多个子图，必须设 `fig.show = 'hold'`，并排图形加起来宽度不能超过文本宽度。如两个子图并排，每个不能超过 50%。

An example of two plots, each with a width of 45% (图4)。

```
par(mar = c(4, 4, .1, .1))
plot(pressure, pch = 19, type = 'b')
plot(cars, pch = 19)
```

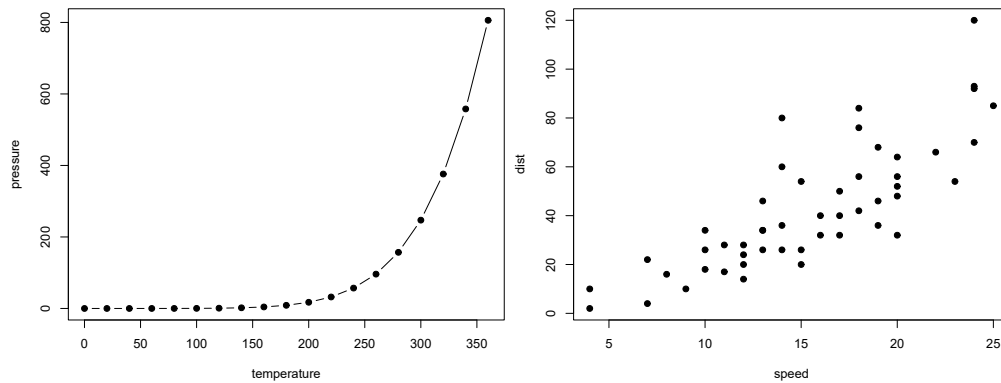


图 4: Two plots placed side by side.

6.4.2 插入图形

R Markdown 用函数 `knitr::include_graphics()` 插入图形，要指明图形文件路径 (图5)。

```
knitr::include_graphics('images/run.jpg')
```



图 5: image included in the document from an external PNG file.

插入图片可以用 `include_graphics()`，也可以用 Markdown 的 ``。

用 `include_graphics()` 插入图形有几点优点：

1. You do not need to worry about the document output format, e.g., when the output format is LaTeX, you may have to use the LaTeX command `\includegraphics{}` to include an image, and when the

output format is Markdown, you have to use ``. The function `include_graphics()` in **knitr** takes care of these details automatically.

2. The syntax for controlling the image attributes is the same as when images are generated from R code, e.g., chunk options `fig.cap`, `out.width`, and `fig.show` still have the same meanings.
3. `include_graphics()` can be smart enough to use PDF graphics automatically when the output format is LaTeX and the PDF graphics files exist, e.g., an image path `foo/bar.png` can be automatically replaced with `foo/bar.pdf` if the latter exists. PDF images often have better qualities than raster images in LaTeX/PDF output. To make use of this feature, set the argument `auto_pdf = TRUE`, or set the global option `options(knitr.graphics.auto_pdf = TRUE)` to enable this feature globally in an R session.
4. You can easily scale these images proportionally using the same ratio. This can be done via the `dpi` argument (dots per inch), which takes the value from the chunk option `dpi` by default. If it is a numeric value and the chunk option `out.width` is not set, the output width of an image will be its actual width (in pixels) divided by `dpi`, and the unit will be inches. For example, for an image with the size 672 x 480, its output width will be 7 inches (7in) when `dpi = 96`. This feature requires the package **png** and/or **jpeg** to be installed. You can always override the automatic calculation of width in inches by providing a non-NULL value to the chunk option `out.width`, or use `include_graphics(dpi = NA)`.

6.5 交叉引用

Rmarkdown 没有交叉引用 (cross-reference) 功能, 但是 bookdown 可以。交叉引用包括数学公式、定理、图形、表格和章节。

用法:

- 标记: 公式 (`\#eq:label`), 图形 (`\#fig:label`), 表格 (`\#tab:label`), 章节 `{#label}` 等。
- 引用: `\@ref(eq:label)`, `\@ref(fig:label)`, `\@ref(tab:label)`, 章节 `\@ref(label)`。

6.5.1 数学公式

要给公式编号, 先要建立公式环境, 然后标记 (`\#eq:label`), 例如

```
\begin{equation}
f\left(k\right) = \text{binom}{n}{k} p^k\left(1-p\right)^{n-k}
(\#eq:binom)
\end{equation}
```

结果如下:

$$f(k) = \binom{n}{k} p^k (1-p)^{n-k} \quad (1)$$

你可以用 `\@ref(eq:binom)` 引用, 如, 参看公式 (1)。

如果公式不需要自动编号, 可建立 `equation*` 环境:

```
\begin{equation*}
\frac{d}{dx}\left(\int_a^x f(u)\,du\right)=f(x)
\end{equation*}
```

$$\frac{d}{dx} \left(\int_a^x f(u) du \right) = f(x)$$

公式对齐: 用 `align`, `=` 号左侧加上对齐标记 `&`; 换行用 `\\`。默认情况下, `align` 环境里每行都会指定一个编号, 如果某行不要编号, 可用 `\notag`。

```

\begin{align}
g(X_{\{n\}}) &= g(\theta) + g'(\tilde{\theta})(X_{\{n\}} - \theta) \quad \text{\notag} \\
\sqrt{n}[g(X_{\{n\}}) - g(\theta)] &= g'(\tilde{\theta})\sqrt{n}[X_{\{n\}} - \theta] \quad (\text{\#eq:align})
\end{align}

```

$$\begin{aligned}
g(X_n) &= g(\theta) + g'(\tilde{\theta})(X_n - \theta) \\
\sqrt{n}[g(X_n) - g(\theta)] &= g'(\tilde{\theta})\sqrt{n}[X_n - \theta]
\end{aligned} \tag{2}$$

如果一个公式有多行，希望共享一个编号，可用 `split` 环境：

```

\begin{equation}
\begin{split}
\mathrm{Var}(\hat{\beta}) &= \mathrm{Var}((X'X)^{-1}X'y) \\
&= (X'X)^{-1}X'\mathrm{Var}(y)((X'X)^{-1}X')' \\
&= (X'X)^{-1}X'\mathrm{Var}(y)X(X'X)^{-1} \\
&= (X'X)^{-1}X'\sigma^2IX(X'X)^{-1} \\
&= (X'X)^{-1}\sigma^2
\end{split}
\end{equation}
(\text{\#eq:var-beta})

```

$$\begin{aligned}
\mathrm{Var}(\hat{\beta}) &= \mathrm{Var}((X'X)^{-1}X'y) \\
&= (X'X)^{-1}X'\mathrm{Var}(y)((X'X)^{-1}X')' \\
&= (X'X)^{-1}X'\mathrm{Var}(y)X(X'X)^{-1} \\
&= (X'X)^{-1}X'\sigma^2IX(X'X)^{-1} \\
&= (X'X)^{-1}\sigma^2
\end{aligned} \tag{3}$$

6.5.2 定理

用下面形式创建一个定理环境：

```

\begin{theorem, label="yourlabel", name="theorem name"}
Here is my theorem.
\end{theorem}

```

一个定理的例子：

```

\begin{theorem, pyth, name="Pythagorean theorem"}
For a right triangle, if $c$ denotes the length of the hypotenuse
and $a$ and $b$ denote the lengths of the other two sides, we have

$a^2 + b^2 = c^2$
\end{theorem}

```

结果如下：

定理 6.1 (Pythagorean theorem) *For a right triangle, if c denotes the length of the hypotenuse and a and b denote the lengths of the other two sides, we have*

$$a^2 + b^2 = c^2$$

引用用法：在需要引用处插入\@ref(thm:label)。

参看定理 6.1。见定义1。

定义、推论、命题、公理、假设等类似。

注意，如果 echo 设置为 FALSE，定理编号失效。

定义 1 随机变量 X 的特征函数定义为

$$\varphi_X(t) = \mathbb{E}[e^{itX}], \quad t \in \mathcal{R}$$

例 1 We derive the characteristic function of $X \sim U(0, 1)$ with the probability density function $f(x) = \mathbf{1}_{x \in [0, 1]}$.

$$\begin{aligned} \varphi_X(t) &= \mathbb{E}[e^{itX}] \\ &= \int e^{itx} f(x) dx \\ &= \int_0^1 e^{itx} dx \\ &= \int_0^1 (\cos(tx) + i \sin(tx)) dx \\ &= \left(\frac{\sin(tx)}{t} - i \frac{\cos(tx)}{t} \right) \Big|_0^1 \\ &= \frac{\sin(t)}{t} - i \left(\frac{\cos(t) - 1}{t} \right) \\ &= \frac{i \sin(t)}{it} + \frac{\cos(t) - 1}{it} \\ &= \frac{e^{it} - 1}{it} \end{aligned}$$

Note that we used the fact $e^{ix} = \cos(x) + i \sin(x)$ twice.

引理 1 For any two random variables X_1, X_2 , they both have the same probability distribution if and only if

$$\varphi_{X_1}(t) = \varphi_{X_2}(t)$$

定理 6.2 If X_1, \dots, X_n are independent random variables, and a_1, \dots, a_n are some constants, then the characteristic function of the linear combination $S_n = \sum_{i=1}^n a_i X_i$ is

$$\varphi_{S_n}(t) = \prod_{i=1}^n \varphi_{X_i}(a_i t) = \varphi_{X_1}(a_1 t) \cdots \varphi_{X_n}(a_n t)$$

命题 1 The distribution of the sum of independent Poisson random variables $X_i \sim \text{Pois}(\lambda_i)$, $i = 1, 2, \dots, n$ is $\text{Pois}(\sum_{i=1}^n \lambda_i)$.

This is the characteristic function of a Poisson random variable with the parameter $\lambda = \sum_{i=1}^n \lambda_i$. From Lemma 1, we know the distribution of P_n is $\text{Pois}(\sum_{i=1}^n \lambda_i)$.

注解 1 In some cases, it is very convenient and easy to figure out the distribution of the sum of independent random variables using characteristic functions.

推论 1 The characteristic function of the sum of two independent random variables X_1 and X_2 is the product of characteristic functions of X_1 and X_2 , i.e.,

$$\varphi_{X_1+X_2}(t) = \varphi_{X_1}(t) \varphi_{X_2}(t)$$

6.6 一个 ggplot2 作图的例子

ggplot2 的几个基本概念：

- 数据 (Data) 和映射 (Mapping)：将数据映到图像
- 几何对象 (Geometric)：代表在图中看到的实际元素，如点、线、多边形等
- 统计变换 (Statistics)：对数据进行某种汇总，如直方图，或将二维关系用线性模型解释
- 标度 (Scale)：将数据的取值映射到图形空间，例如用：颜色、大小、形状表示不同取值
- 坐标系 (Coordinate)：数据如何映射到图形所在平面，提供作图所需的坐标轴和网格线
- 分面 (Facet)：将数据分解为子集，进行联合展示
- 图层 (Layer)：对所需的绘图操作进行一层一层叠加，最终得到所需图形

6.6.1 散点图

```
library(ggplot2)
p <- ggplot(data = mpg, mapping = aes(x = cty, y = hwy))
p + geom_point()
```

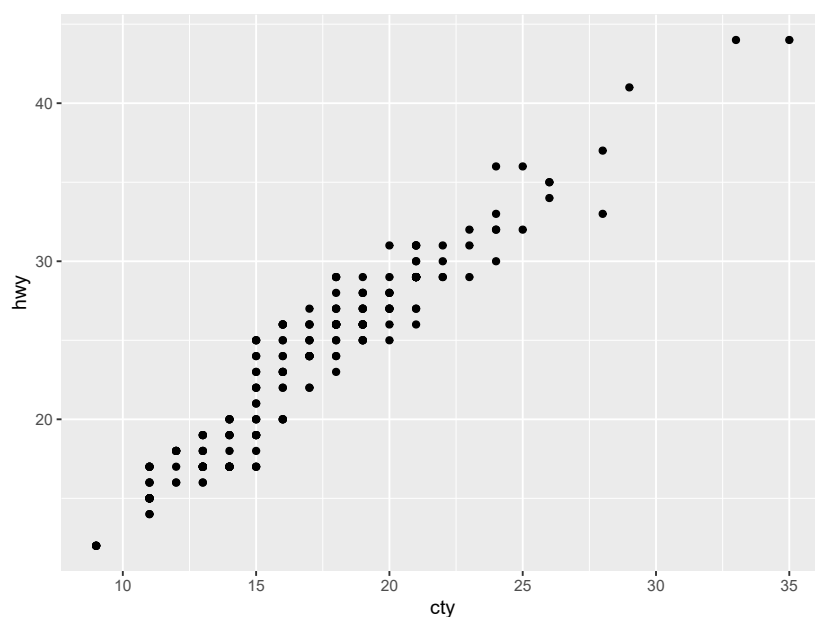


图 6: ggplot2 with points

第一行指定数据集、映射（坐标轴），第二行表示在 `p` 的基础上加上点，`geom` 表示的是 `geometric object`（几何对象）（见图6）。

`mpg` 是 `ggplot2` 里面的一个关于汽车的数据集。

- `cty`: city miles per gallon
- `hwy`: highway miles per gallon
- `year`: year of manufacture
- `displ`: engine displacement, in litres
- `cyl`: number of cylinders
- `class`: “type” of car

6.6.2 变换颜色

按生产年份以颜色区分，`factor(year)` 是把年份转化为因子形式（相当于定类变量），见图 7。

```
p <- ggplot(mpg, aes(x = cty, y = hwy, colour = factor(year)))
p + geom_point()
```

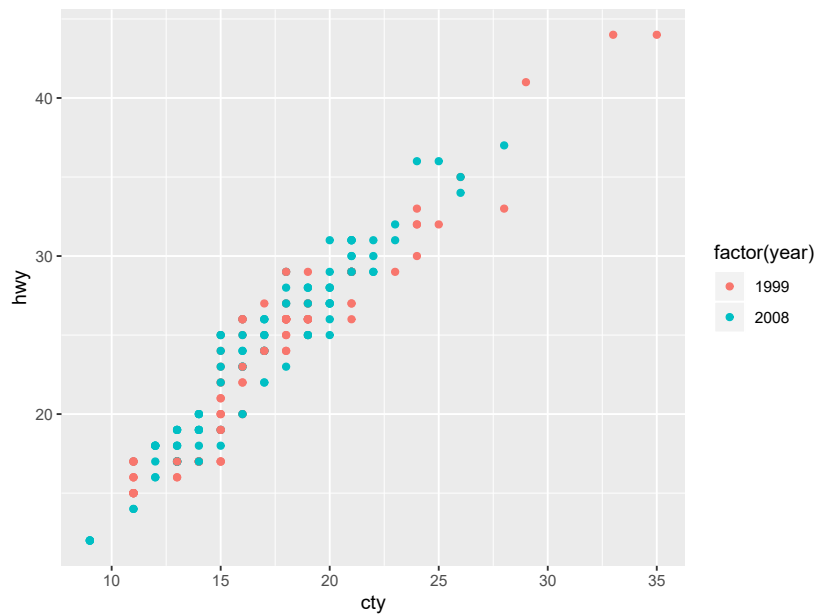


图 7: ggplot2 with points colors

6.6.3 拟合曲线

再加一行 `+ stat_smooth()`，其中 `stat` 表示 statistical transformation，做了统计平滑拟合直线，以及置信区间，见图 8。

```
p + geom_point() + stat_smooth()
```

6.6.4 变换大小

上图的数据点明显偏小，可以让这些数据点随着汽车排量的大小而变化，见图 9。

```
p + geom_point(aes(colour = factor(year), size = displ)) +
  stat_smooth() # 排量越大，点越大
```

6.6.5 修改透明度

数据点太密集，增加透明度，解决点与点之间的重叠的问题，见图 10。

```
p + geom_point(aes(colour = factor(year),
                    size = displ), alpha = 0.5) +
  stat_smooth() + scale_size_continuous(range = c(4, 10))
```

`alpha=0.5` 在 `aes()` 的外面，代表对所有的点都强制透明度为 0.5。

6.6.6 图形分层

1999 年与 2008 年数据点全部挤在一块，太拥挤了，应采用分层，见图 11。

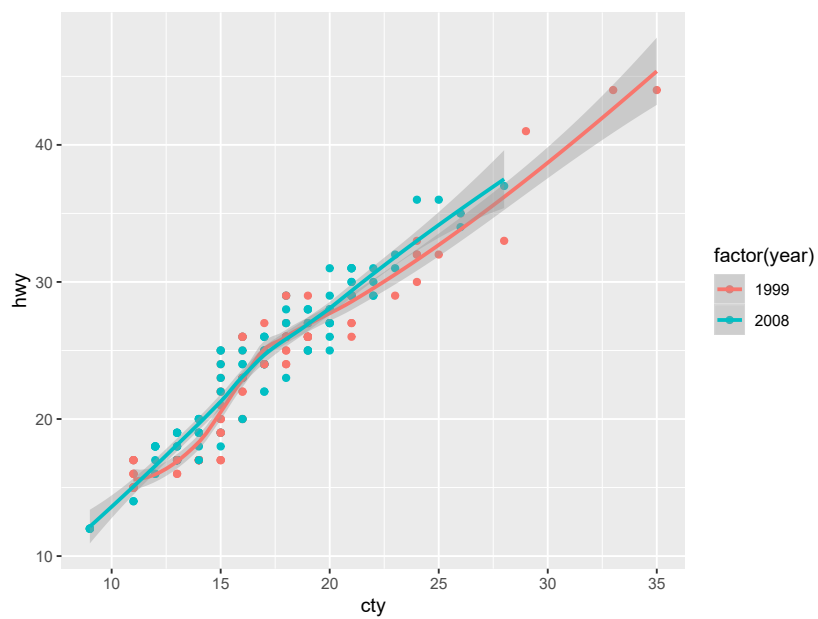


图 8: ggplot2 with smooth trends

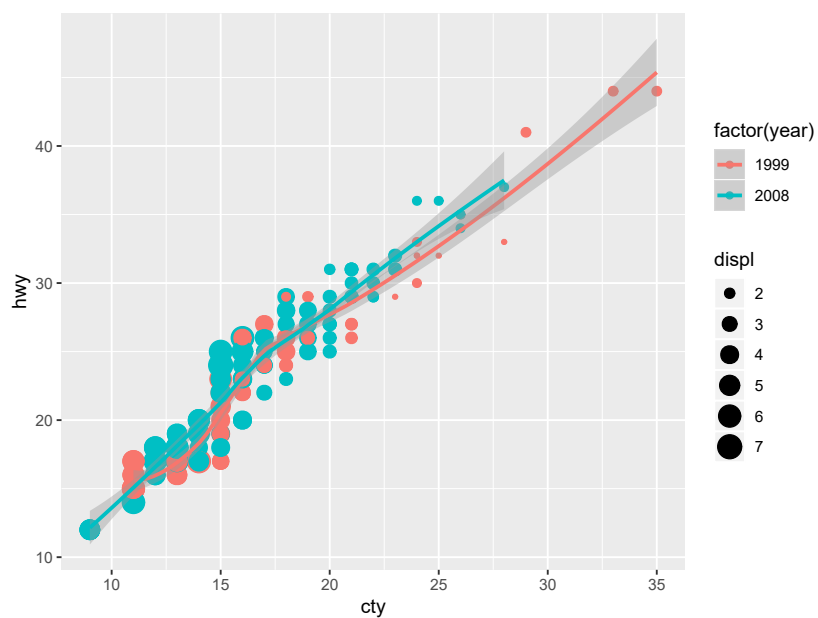


图 9: ggplot2 with variable point size

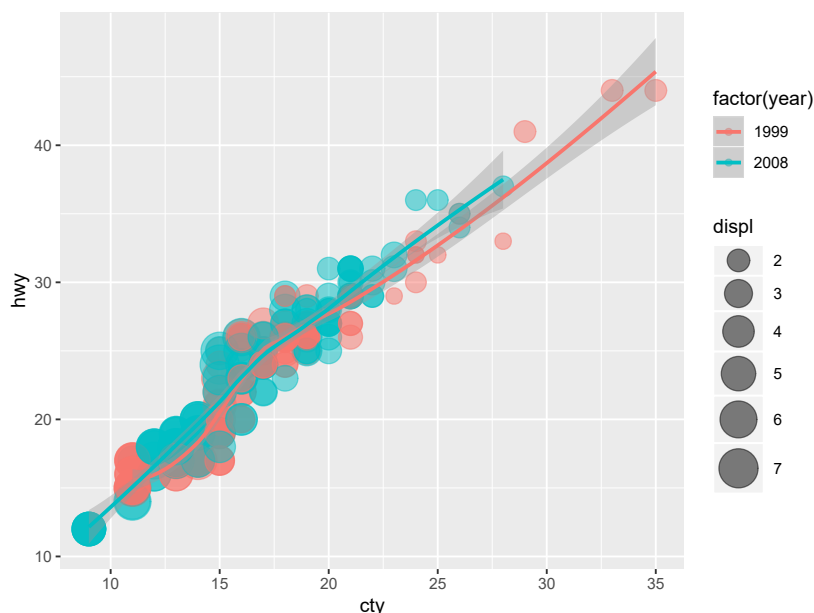


图 10: ggplot2 with transparent points

```
p + geom_point(aes(colour = class, size = displ), alpha = 0.5) +
  stat_smooth() + scale_size_continuous(range = c(4, 10)) +
  facet_wrap(~ year, ncol = 1)
```

- `facet_wrap()` 是 `facet` 与 `wrap` 两个词组合，表示逐面包起来。
- `~year` 表示按变量 `year` 分层，将 1999 与 2008 分开。
- `ncol = 1` 代表小窗口是 1 列，指定了 1 列之后，默认就是两行（因为年份一共只有两种）。如果不加这句，会默认横着排列，或者想要指定几行，则使用 `nrow = 1`。
- 这里颜色指定了 `colour = class`，代表不同种类的汽车。
- 添加了 `scale_size_continuous(range = c(4, 10))`，指定 `size` 的变化范围。在本图中，就是控制点的绝对大小的范围，不要太大，也不要太小。

6.6.7 添加中文标注

默认情况下图形上是不能出现中文的，要使得中文在图形中正常显示，必须在文档开头的 `output` 下面加上：`dev: "cairo_pdf"`。也可以在作图时加上 `pdf.options(family="GB1")`。

```
p + geom_point(aes(colour = class, size = displ), alpha=0.5) +
  stat_smooth() + scale_size_continuous(range = c(4, 10)) +
  facet_wrap(~ year, ncol = 1) +
  labs(y = ' 每加仑高速公路行驶距离', x = ' 每加仑城市公路行驶距离',
       title = ' 汽车油耗与型号', size = ' 排量', colour = ' 车型') +
  theme(text = element_text(family = "STHeiti"),
        plot.title = element_text(hjust = 0.5))
```

图 12 中，

- `labs()` 修改的是标签的名称。
- `theme()` 主题，更偏向于格式的修改。`text = element_text(family = "STHeiti")` 是对字体进行修改，变为黑体。Windows 系统可以不添加这行，一样会显示前面 `labs()` 中设定的中文。而如果是 Mac 或者 Linux 系统，由于字体的缺失，会显示成一个一个的框框，在图像上显示不了中文字。
- `plot.title = element_text(hjust = 0.5)` 调整标题的位置，不加这行，标题会居左，加上才会居中。`hjust = 0.5` 其实就是左右移动的意思，0.5 表示居中。

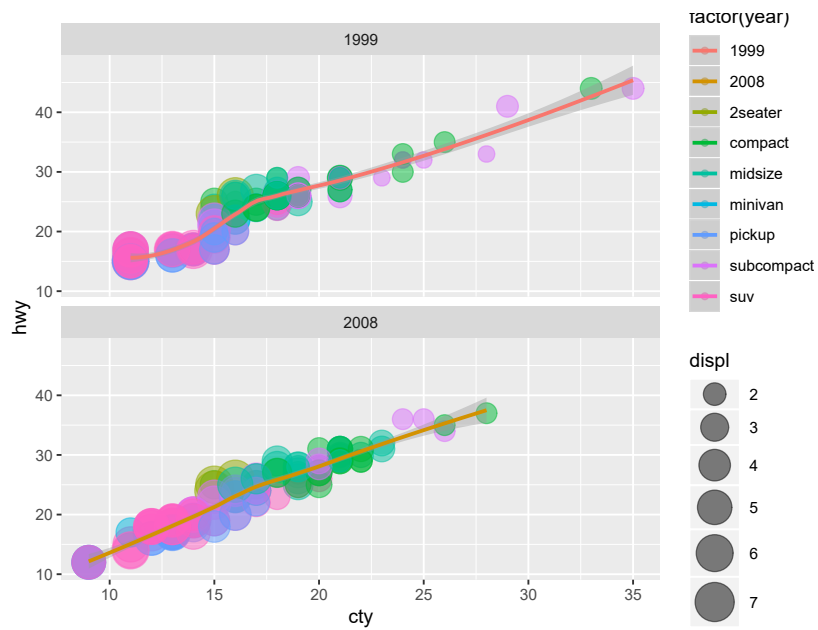


图 11: ggplot2 with facets

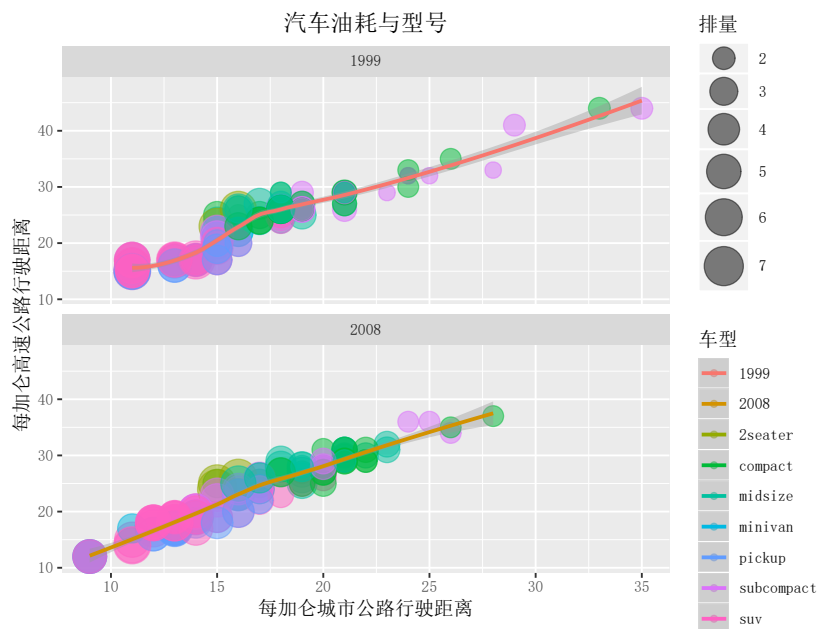


图 12: ggplot2 with Chinese characters

表 1: A table of the first 10 rows of the mtcars data.

	mpg	cyl	disp	hp	drat	wt	qsec	vs
Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0
Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1
Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0
Valiant	18.1	6	225.0	105	2.76	3.460	20.22	1
Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0
Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1
Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1
Merc 280	19.2	6	167.6	123	3.92	3.440	18.30	1

表 2: A Tale of Two Tables.

Sepal.Length	Sepal.Width		mpg	cyl	disp
5.1	3.5	Mazda RX4	21.0	6	160
4.9	3.0	Mazda RX4 Wag	21.0	6	160
4.7	3.2	Datsun 710	22.8	4	108
		Hornet 4 Drive	21.4	6	258
		Hornet Sportabout	18.7	8	360

这个例子来自 https://blog.csdn.net/weixin_41929524/article/details/79765882。

6.7 表格

6.7.1 用 R 函数 `kable()` 制作表格

R Markdown 对数据制作表格最方便的函数是 `knitr::kable()`。

```
knitr::kable(
  head(mtcars[, 1:8], 10), booktabs = TRUE,
  caption = 'A table of the first 10 rows of the mtcars data.'
)
```

If you want to put multiple tables in a single table environment, wrap the data objects (usually data frames in R) into a list. Please note that this feature is only available in HTML and PDF output.

```
knitr::kable(
  list(
    head(iris[, 1:2], 3),
    head(mtcars[, 1:3], 5)
  ),
  caption = 'A Tale of Two Tables.', booktabs = TRUE
)
```

在 LaTeX 的 pdf 中，插图经常会浮动位置。When you do not want a table to float in PDF, you may use the LaTeX package **longtable**, which can break a table across multiple pages. To use **longtable**, pass `longtable = TRUE` to `kable()`, and make sure to include `\usepackage{longtable}` in the LaTeX preamble for how to customize the LaTeX preamble). Of course, this is irrelevant to HTML output, since tables in HTML do not need to float.

```
knitr::kable(
  iris[1:5, ], longtable = TRUE, booktabs = TRUE,
  caption = 'A table generated by the longtable package.'
)
```

表 3: A table generated by the longtable package.

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa

6.7.2 制作 Markdown 表格

用 Markdown 制作表格很简单，直接用--- 和 | 分隔即可。

Pandoc supports several types of Markdown tables, such as simple tables, multiline tables, grid tables, and pipe tables. What `knitr::kable()` generates is a simple table like this:

Table: A simple table in Markdown.

First Header	Second Header
Content Cell	Content Cell
Content Cell	Content Cell

表 4: A simple table in Markdown.

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
5.1	3.5	1.4	0.2
4.9	3.0	1.4	0.2
4.7	3.2	1.3	0.2
4.6	3.1	1.5	0.2
5.0	3.6	1.4	0.2
5.4	3.9	1.7	0.4

You can use any types of Markdown tables in your document.

6.8 文献引用

学术论文一般做法是，把所有文献以 **BibTeX** 格式保存为一个 .bib 文件，然后在论文中随时插入引用。

引用：如果是作者-年格式，`@R-rmarkdown` 表示作者姓名（年份），而 `[@R-rmarkdown]` 则表示（作者年份）。

如果是编号格式，则 `@R-rmarkdown` 表示序号，而 `[@R-rmarkdown]` 则表示序号为上标。

示例 Allaire et al. (2018); 示例 [Allaire et al., 2018]。

如何产生 Bibtex 文献格式？

- 一般用文献管理软件（如 Zotero、Endnote 等）把参考文献转为 .bib 文件。

- 可以在知网、谷歌学术、百度学术等网站查找文献，产生 Bibtex 引用格式。
- 一些常用软件包夜可以用 **knitr** 的函数 `write_bib()` 产生。

Although Pandoc supports multiple ways of writing citations, we recommend you to use **BibTeX** databases because they work best with LaTeX/PDF output. Pandoc can process other types of bibliography databases with the utility `pandoc-citeproc` (<https://github.com/jgm/pandoc-citeproc>), but it may not render certain bibliography items correctly (especially in case of multiple authors per item), and BibTeX can do a better job when the output format is LaTeX. With BibTeX databases, you will be able to define the bibliography style if it is required by a certain publisher or journal.

A BibTeX database is a plain-text file (with the conventional filename extension `.bib`) that consists of bibliography entries like this:

```
@Manual{R-base,
  title = {R: A Language and Environment for Statistical
    Computing},
  author = {{R Core Team}},
  organization = {R Foundation for Statistical Computing},
  address = {Vienna, Austria},
  year = {2016},
  url = {https://www.R-project.org/},
}
```

A bibliography entry starts with `@type{}`, where `type` may be `article`, `book`, `manual`, and so on.² Then there is a citation key, like `R-base` in the above example. To cite an entry, use `@key` or `[@key]` (the latter puts the citation in braces), e.g., `@R-base` is rendered as `R Core Team (2018)`, and `[@R-base]` generates “(R Core Team, 2018)”. If you are familiar with the **natbib** package in LaTeX, `@key` is basically `\citet{key}`, and `[@key]` is equivalent to `\citep{key}`.

There are a number of fields in a bibliography entry, such as `title`, `author`, and `year`, etc. You may see <https://en.wikipedia.org/wiki/BibTeX> for possible types of entries and fields in BibTeX.

There is a helper function `write_bib()` in **knitr** to generate BibTeX entries automatically for R packages. Note that it only generates one BibTeX entry for the package itself at the moment, whereas a package may contain multiple entries in the CITATION file, and some entries are about the publications related to the package. These entries are ignored by `write_bib()`.

```
# the second argument can be a .bib file
knitr::write_bib(c('knitr', 'bookdown'), '', width = 60)
```

```
@Manual{R-bookdown,
  title = {bookdown: Authoring Books and Technical Documents
    with R Markdown},
  author = {Yihui Xie},
  year = {2018},
  note = {R package version 0.7},
  url = {https://CRAN.R-project.org/package=bookdown},
}

@Manual{R-knitr,
  title = {knitr: A General-Purpose Package for Dynamic Report
    Generation in R},
  author = {Yihui Xie},
  year = {2018},
  note = {R package version 1.20},
  url = {https://CRAN.R-project.org/package=knitr},
}
```

Once you have one or multiple `.bib` files, you may use the field `bibliography` in the YAML metadata

²The type name is case-insensitive, so it does not matter if it is `manual`, `Manual`, or `MANUAL`.

of your first R Markdown document (which is typically `index.Rmd`), and you can also specify the bibliography style via `biblio-style` (this only applies to PDF output), e.g.,

```
---
bibliography: ["one.bib", "another.bib", "yet-another.bib"]
biblio-style: "apalike"
link-citations: true
---
```

The field `link-citations` can be used to add internal links from the citation text of the author-year style to the bibliography entry in the HTML output.

When the output format is LaTeX, citations will be automatically put in a chapter or section. For non-LaTeX output, you can add an empty chapter as the last chapter of your book. For example, if your last chapter is the Rmd file `06-references.Rmd`, its content can be an inline R expression:

```
`r if (knitr::is_html_output()) '# References {-}'`
```

为了试试中文的引用，我通过 Zotero 产生一个 bibtex 文献（注意中文引用显示为—，需要自行修改）（柯忠义, 2017）。

参考文献

Allaire, J., Xie, Y., McPherson, J., Luraschi, J., Ushey, K., Atkins, A., Wickham, H., Cheng, J., and Chang, W. (2018). *rmarkdown: Dynamic Documents for R*. R package version 1.10.

R Core Team (2018). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Xie, Y. (2018a). *bookdown: Authoring Books and Technical Documents with R Markdown*. R package version 0.7.

Xie, Y. (2018b). *knitr: A General-Purpose Package for Dynamic Report Generation in R*. R package version 1.20.

Zhao, P. (2017). *bookdownplus: Generate Varied Books and Documents with R 'bookdown' Package*. R package version 1.0.2.

柯忠义 (2017). 创业板上市公司经济绩效及影响因素——基于贝叶斯模型平均法 (BMA) 的实证研究. 数量经济技术经济研究, 34(1):146–160.