

如何用 R Markdown 写学术文档?

王树佳
深圳大学经济学院
wangsj@szu.edu.cn

2019 年 09 月 14 日

目录

1	介绍	3
1.1	什么是 R Markdown?	3
1.2	搭建系统环境	4
2	用 R Markdown 写简单文档	5
3	用 R Markdown 写学术文档	6
4	用 Bookdown 写书或长文	8
4.1	用 bookdown 写中英文书籍	8
4.2	用 Bookdownplus 模版快速上手	8
4.2.1	安装调用	8
4.2.2	用 Bookdownplus 撰写书籍	8
5	修改中文参考文献样式	9
5.1	一般学术文档修改	9
5.2	修改 Bookdownplus	9
5.3	修改 Bookdown 任意样式	9
6	R Markdown 简介	9
6.1	Markdown 基本语法	9
6.1.1	段落	9
6.1.2	段内文字格式	9
6.1.3	标题和分隔线	10
6.1.4	引用段落	10
6.1.5	列表	10
6.1.6	源程序代码	10
6.1.7	链接	11
6.2	数学公式	11
6.2.1	Markdown 公式	11
6.2.2	LaTeX 公式环境	13
6.2.3	参考手册	14
6.3	R 代码块	14
6.4	图形	15
6.4.1	插入图片	15
6.4.2	R 做图	15
6.4.3	一个 ggplot2 作图的例子	17
6.5	表格	21
6.5.1	用 R Markdown 函数 <code>kable()</code> 制作表格	21
6.5.2	用 R 软件包 <code>xtable</code> 和 <code>stargazer</code> 制作表格	23
6.5.3	Markdown 表格	23

6.6	交叉引用	24
6.6.1	数学公式	24
6.6.2	定理	25
6.6.3	交互引用方式列表	26
6.7	文献引用	27
7	创建 Git 版本跟踪项目	27
8	总结	28

1 介绍

本文的主要目的是介绍如何用 R Markdown 撰写学术类型的文档。

主要内容包括：

- (一) 简单介绍 R Markdown 及需安装的软件和设置；
- (二) 简单的学术类文档（结构简单，不需要交叉引用，如理工科课程作业等）；
- (三) 完整的学术类文档（结构完整，篇幅不长，但需要交叉引用，如学术论文等）¹；
- (四) 撰写长篇幅的学术类文档（全文需分章节，篇幅较长，需要跨章节的交叉引用，如书籍、学位论文等）；
- (五) 中文参考文献的样式；
- (六) 介绍 R Markdown 的具体用法。

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。

R Markdown 使得数据分析代码可以与文档混编，具体语法请参看第6节。

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：

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

¹ 本文就是这种类型的文档

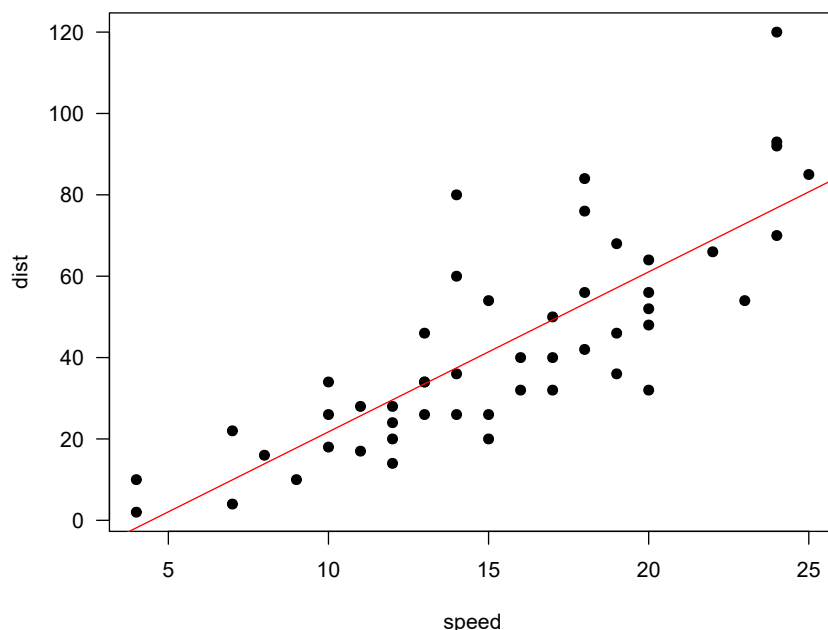


图 1: cars 数据散点图以及回归直线。

1.2 搭建系统环境

首先需要下载并安装 R、RStudio 和 Texlive 等软件。

(一) 下载并安装 R

到如下网址下载最新版 R 并安装: <https://www.r-project.org/>。一般默认安装即可, 不过, 根据 Xie Yihui 的经验, Windows 中安装 R 如下注意几点会更好²:

- 把默认安装目录中的 R 版本号去掉, 比如 R 默认装在 C:/Program Files/R/R-3.5.3/, 不要接受这个结果, 把它改成 C:/Program Files/R/, 以后升级软件仍然安装在这个文件夹;
- 不能安装在中文目录下;
- 用自定义方式安装。不要安装翻译文件。
- 配置 .Renvirom 文件。这个文件是为 R 自身设置一些环境变量的, 这里面的环境变量仅仅对 R 有效, 不改变操作系统的设置。 .Renvirom 文件放在用户目录下, 也就是 ~/. 如果文件不存在, 可以用如下命令自动创建: `file.edit('~/.Renvirom')`。这个文件可以直接打开编辑, 推荐放一个环境变量在里面, 就是 `R_LIBS_USER=~/.R`, 它指定 R 的软件包安装目录在 ~/.R。重启 R, 以后安装 R 包就优先安装到 ~/.R 目录下。因为这个目录是固定的, 所以即使将来升级 R 版本, R 也会使用这个目录装包, 以后再也不必更新关于包的配置。

(二) 下载并安装 RStudio

到如下网址下载最新版 RStudioR 并安装: <https://www.rstudio.com/>。

RStudio 的基本设置

首先对 RStudio 进行正确的设置:

- 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

至此, 已经可以完美运行 R 和纯英文的 RMarkdown 了。不过, 如果要用中文撰写 RMarkdown 文档, 则需要安装 LaTeX。

²<https://bookdown.org/yihui/r-ninja/setup.html#configure-r>

最好每项工作都单独建立一个文件夹，把性质相同的文档归到同一个子目录（比如要插入的图形、需调用的参考文献、设置的格式和参数等等）。然后建立 RStudio 项目 (project)，每次开始都打开这个 project，这样项目的一切材料都会齐备，RStudio 的工作目录不会搞乱。

（三）下载并安装 Texlive2019

到如下网址下载 Texlive2019: <https://mirrors.tuna.tsinghua.edu.cn/CTAN/systems/texlive/Images/>，下载文件 texlive2019-20190410.iso (3.3G)，用解压缩软件 Winrar 解压，运行 install-tl-windows.bat 默认安装。软件很大，一般电脑安装时间要一个多小时。不过文件很齐全，安装后，就可以轻松驾驭著名的科技论文写作神器 LaTeX 了。也可以安装 Latex 前端 LyX (<https://www.lyx.org>)，写 paper 非常好用。

有时用 xelatex 编译会很慢，往往是中文字体缓存刷新问题。解决办法：

1. 清空 C:\texlive\2019\texmf-var\fonts\cache 中的文件；
2. 用管理员命令运行 `fc-cache -r -v` 重新生成字体缓存。

（四）安装常用软件包

首先安装软件包 bookdown(Xie 2018a) 及其相依软件包，R Markdown 写作所需的包基本上就齐了。bookdown 包括了 rmarkdown (Allaire et al. 2018)、knitr (Xie 2018b)、rticles 等，如果有包缺失，运行时提示安装。

bookdown 网站: <https://bookdown.org/>，可以安装稳定版: `install.packages('bookdown')`，也可以安装开发版: `devtools::install_github('rstudio/bookdown')`

其次建议安装软件包 tidyverse，它包含了数据分析常用的几个核心包，包括：

- ggplot2 数据可视化
- dplyr 数据变换和操作
- tidyr 数据整理
- readr 数据输入
- purrr 函数编程
- tibble 新的数据框格式
- stringr 简单易用的字符串工具集
- forcats 用于处理分类变量的因子 (Factor)

2 用 R Markdown 写简单文档

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 R Markdown 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 knitr 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。注意：pdf 文档中不能出现任何中文字符，包括标点符号。
 - 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，需要加装软件包并正确设置。

一般文档开头（称为 yaml）是这样的：

```

---
title: "Title"
author: "Your Name"
date: "2019-09-14"
output:
  html_document:
    fig_caption: yes
    number_sections: yes
    toc: yes
    toc_depth: 3
    toc_float: True
  pdf_document:
    fig_caption: yes
    keep_tex: yes
    latex_engine: xelatex
    number_sections: yes
    toc: yes
  word_document:
    fig_caption: yes
    toc: yes
bibliography: reference.bib
bibli0-style: apalike
link-citations: yes
colorlinks: yes
#lot: yes
#lof: yes
---

```

中文 pdf 文档

首先要确保已经安装 `bookdown`，因为中文 pdf 文档需要用到 `rticles`, `tynytex` 等软件包。中文 pdf 文档还需要用到 LaTeX 套件，如 `Texlive`, `MikTeX` 等。安装 `bookdown` 后，这些套件是否需要安装，我不能确定。

创建简单的中文 pdf 文档，很简单：打开 R Markdown 时，选择 `From Template`，从模版中选择 `CTex Documents` 即可。

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不能交互引用。

为了实现交叉引用功能，可以扩展到 `bookdown`。但是 `bookdown` 需要建立一个文件夹，里面有一堆文件夹和文件，还要每章（节）单独写一个 Rmd 文件，有点复杂。除非你写书或硕博毕业论文，一般不太长的文档或杂志论文没有必要。解决办法是把 `bookdown` 用单个文件的格式 (Xie 2018a)，也具有交互引用功能。

一个合适的设置需要满足：

- (1) 标题、图形、表格、公式等可以自动编号，可以交互引用；
- (2) 一个文档可以输出 `html`、`pdf` 和 `word` 等多种格式；
- (3) 参考文献可以方便地修改样式。

我的 `yaml` 是这样的（满足以上三个条件）：

```

---
title: " 如何用 R Markdown 写学术文档？ "
author: "Shujia Wong"
#date: "2019-09-14"
date: "2019 年 09 月 14 日"

```

```

#geometry: left = 3.18cm, right = 3.18cm, top = 2.0cm, bottom = 2.0cm
#fontsize: 12pt
output:
  bookdown::gitbook:
    config:
      toc:
        collapse: subsection
      self.contained: yes
      table_css: TRUE
      number_sections: TRUE
      split_by: none
  bookdown::pdf_document2:
    keep_tex: yes
    dev: "cairo_pdf" # Allow Chinese figure title and labels
    latex_engine: xelatex # for chinese
    template: template/template_article_zh.tex #latex template
    #pandoc_args: --top-level-division=chapter
    fig_caption: yes
    number_sections: yes
    toc: yes
    toc_depth: 3
    #toc_unnumbered: no
    #toc_appendix: yes
    quote_footer: ["\\begin{flushright}", "\\end{flushright}"]
  bookdown::html_document2:
    css: bib/style.css
    fig_caption: yes
    theme: readable
    #highlight: monochrome
    number_sections: yes
    toc: yes
    toc_depth: 3
    toc_float: True # 目录作为侧边栏
  bookdown::word_document2:
    reference_docx: template/reference_word.docx
    fig_caption: yes
    toc: yes
    toc_depth: 3
citation_package: natbib
bibliography: [bib/packages.bib,bib/book.bib]
#csl: bib/chinese-gb7714-1987-numeric.csl
bibli0-style: apalike
link-citations: yes
colorlinks: yes
#lot: yes
#lof: yes
---
```

注意，要在工作目录中创建两个子目录：

- `template` 子目录，存放样板文档。`template_article_zh.tex` 来自 `bookdownplus` (Zhao 2017)，并且稍作修改。`reference_word.docx` 是本人制作，你可以根据需要修改 `word` 的样式。
- `bib` 子目录，存放参考文献（`.bib`）和式样文件（`.csl`）。

4 用 Bookdown 写书或长文

4.1 用 bookdown 写中英文书籍

长篇文章（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 R Markdown 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 (Xie 2018a)。

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

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。

中文可输出 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/>

4.2 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。

4.2.1 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: `get_template()`，其中 `article` 是英文版学术论文模版，`article_zh` 是中文学术论文模版，支持双语标题、作者、摘要、关键词。`pku_zh` 是北京大学毕业论文模版，`ucas_zh` 是中国科学院大学学位论文模版，`zju_zh` 是浙江大学毕业论文模版。

4.2.2 用 Bookdownplus 撰写书籍

步骤：

- (1) 产生模版。首先把 RStudio 的工作目录设置在一个空白的文件夹 (Session -> Set Working Directory -> Choose Directory，也可以通过快捷键选取 Ctrl + Shift + H)。执行 `library("bookdownplus") bookdownplus(template = "article_zh", rproj=TRUE)` 其中 `rproj=TRUE` 必须加上，否则不会产生 `bookdownplus.Rproj` 文件。
- (2) 在 `index.Rmd` 修改题目、作者等，在 `body.Rmd` 撰写正文。注意这里一级标题竟然要用 `##`，有点不明白。

- (3) 创建你的书籍。在 RStudio 中打开 `bookdownplus.Rproj`，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，你的书籍在文件夹 `_book` 里面。
- 其它格式（书籍、毕业论文、杂志文章等）类似，请自行测试。

5 修改中文参考文献样式

不管是 Bookdown 还是 Bookdownplus，默认参考文献的引用格式都是 `apalike`，引用格式是作者-年，不符合 `gb7714` 中文标准。

如何把参考文献修改为中文标准格式？

5.1 一般学术文档修改

- (1) 到 `csl` 样式库 <https://www.zotero.org/styles> 查看合适的样式（有几百种），可以到 <https://github.com/citation-style-language/styles> 去下载源文件 (Raw)，比如 `chinese-gb7714-1987-numeric.csl`，放在工作目录下的 `bib` 子目录。
- (2) 样式的调用：注释掉 `#bibliography-style: apalike`，然后增加一行即可：
- ```
csl: bib/chinese-gb7714-1987-numeric.csl
```

为了试试中文的引用，我通过 Zotero 产生一个 `bibtex` 格式中文文献（注意自动生成的中文 `key` 显示为--，需要自行修改），例如 (柯忠义2017)。

### 5.2 修改 Bookdownplus

- 把 `chinese-gb7714-1987-numeric.csl` 放在项目所在目录。
- 在文件 `index.rmd` 中：
  - 修改 `natbib` 为 `none`;
  - 把 `pandoc_args:` 后面内容替换为: `[ "--top-level-division=chapter", "--csl", "chinese-gb7714-1987-numeric.csl" ]`;
  - 删除或注释掉文档 `index.Rmd` 中的 `bibliography-style: apalike`;
- 在 `body.Rmd` 的最后增加一行: `## 参考文献 {-}`。

### 5.3 修改 Bookdown 任意样式

- 在文件 `_output.yml` 中修改 `citation_package: none`;
- 在文件 `_output.yml` 各种格式 (`gitbook, pdf_book, epub_book`) 中增加 `pandoc_args:` `[ "--csl", "your-csl-file.csl" ]`;
- 删除或注释掉文档 `index.Rmd` 中的 `bibliography-style: apalike`;
- 把文件 `08-references.Rmd` 首行改为: `# 参考文献 {-}`（否则参考文献不出现）。

## 6 R Markdown 简介

### 6.1 Markdown 基本语法

#### 6.1.1 段落

Markdown 中，一个段落由一行或连续的多行组成。段落之间以空行分隔。同一段落内的不同行仅表示空格。

为了在段内换行，可在前行末尾输入两个或两个以上空格。也可以输入 `<br>` 分行，但是仅对 `html` 和 `word` 格式有效，对 `pdf` 格式无效。

#### 6.1.2 段内文字格式

- 斜体：星号或下划线包围的内容是强调格式，如 `_text_` 或 `*text*`: *text*。
- 黑体：双星号或双下划线包围的内容是加重格式，如 `text`, `__text__` 或 `**text**`。
- 下标：两个 `~` 包夹的内容为下标，如 `H~2~S0~4~` 显示为  $H_2SO_4$ 。

- 上标：两个 `^` 包夹内容为上标，`R^2^` 显示为  $R^2$ 。
- 显示代码：两个反向单撇号内，如 ``code``，显示为：`code`<sup>3</sup>。
- 为了使得某些有特殊意义的字符不作特殊解释，可以在该字符前面加上反斜杠`\`，取消其特殊含义。比如，要插入普通的星号或下划线，可以使用反斜杠保护`\*`，或者写成段内代码格式 ``*``。
- 插入图片：`![图片标题](path/to/image)`，更好的图形插入方式在第6.4节介绍。
- 脚注：`^[脚注文本]`。例如，这里插入脚注<sup>4</sup>

### 6.1.3 标题和分隔线

- 一级标题：以井号 `#` 开头，或在标题下面一行输入多个等号 `=====`；
- 二级标题：两个井号 `##` 开始。或在标题下面一行输入多个减号`-----`；
- 标题行前面应该空一行；
- 用三个或三个以上连续的星号组成的行，可以转换成分隔线；
- 标题不编号：标题后增加 `{-}`，如本文最后标题“`# 参考文献 {-}`”。

### 6.1.4 引用段落

在段落的每行前面加一个大于号 `>`，表示引用。

凡事只要看得淡些，就没有什么可忧虑的了；只要不因愤怒而夸大事态，就没有什么事情值得生气的了。

——屠格涅夫

### 6.1.5 列表

不编号列表：以 `*`、`-` 或 `+` 开头，空两格。下一行再空四个空格可以嵌套，如：

```
- 第一项
- 第二项
 - 第二项的第一项
 - 第二项的第二项
```

结果：

- 第一项
- 第二项
  - 第二项的第一项
  - 第二项的第二项

带编号列表：

```
1. 第一项
2. 第二项
3. 第三项
```

### 6.1.6 源程序代码

三个或以上连续的反向单撇号表示代码开头与代码结束，缩进四个空格也可以。

```
```
这是代码
```
```

或缩进四个空格：

```
 这是代码
```

使用栅栏式代码块，可以彩色语法和行号显示等多种选项，如`.cpp`表示C++，`.c`表示C，`.r`表示R，`.python`表示python等。选项`numberLines`要求该代码行编号，选项`startFrom=`指定开始行号。

<sup>3</sup>要把反向单撇号作为一般字符（而不是代码表示符），可以在外围用比它多的反向单撇号表示，如 ```` `code` ````。

<sup>4</sup>所谓“脚注”，就是在页脚对某些文字的注释。

另外，如果代码内本身含有 ~ 行，只要使得开头与结尾标志中的 ~ 个数更多就可以了。

```
~~~{.python .numberLines startFrom=101}
# 计算面积函数
def area(width, height):
    return width * height

def print_welcome(name):
    print("Welcome", name)

print_welcome("Runoob")
}
~~~
```

结果:

```
101 # 计算面积函数
102 def area(width, height):
103 return width * height
104
105 def print_welcome(name):
106 print("Welcome", name)
107
108 print_welcome("Runoob")
109 }
```

### 6.1.7 链接

- 直接链接。  
深圳大学的网页地址是： <<https://www.szu.edu.cn/>> 。  
显示为深圳大学的网页地址是： <https://www.szu.edu.cn/>。
- 行内式链接。  
[显示的文本](链接网址)，  
例如，请参考 [RStudio](<https://www.rstudio.com>)，  
显示为RStudio。

## 6.2 数学公式

### 6.2.1 Markdown 公式

行内公式用两个美元符号， $y=x^2$  显示  $y = x^2$ 。

整行公式（display style）最简单的方法是用两双美元符号之间表示：

$$f(x) = \frac{1}{2} \sum_{i=1}^{\infty} \int_0^1 \sin^2(itx) dt.$$

$$f(x) = \frac{1}{2} \sum_{i=1}^{\infty} \int_0^1 \sin^2(itx) dt.$$

矩阵和行列式公式

可以通过`\begin{array}`和`\end{array}`：

```
$$X=\left(\begin{array}{ccc}
x_{11} & x_{12} & x_{13} \\
x_{21} & x_{22} & x_{23}
\end{array}\right)$$
```

$$X = \begin{pmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \end{pmatrix}$$

也可以通过`\begin{matrix}`和`\end{matrix}`（有 b, B, p, P, v, V 等多种格式）：

```


$$\begin{matrix} a & b \\ c & d \end{matrix}$$

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$\begin{Bmatrix} a & b \\ c & d \end{Bmatrix}$$

$$\begin{vmatrix} a & b \\ c & d \end{vmatrix}$$

$$\left| \begin{matrix} a & b \\ c & d \end{matrix} \right|$$

$$\begin{Vmatrix} a & b \\ c & d \end{Vmatrix}$$


```

$$\begin{matrix} a & b \end{matrix}$$

$$\begin{matrix} c & d \end{matrix}$$

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$\begin{Bmatrix} a & b \\ c & d \end{Bmatrix}$$

$$\begin{vmatrix} a & b \\ c & d \end{vmatrix}$$

$$\begin{vmatrix} a & b \\ c & d \end{vmatrix}$$

$$\begin{Vmatrix} a & b \\ c & d \end{Vmatrix}$$

分段函数：

```

f(m,n)=
\begin{cases}
x(n) & x>0, \\
0 & x=0, \\
x(n-1) & x<0.
\end{cases}

```

$$f(m,n) = \begin{cases} x(n) & x > 0, \\ 0 & x = 0, \\ x(n-1) & x < 0. \end{cases}$$

这样的整行公式输出 html 格式非常简单，效果也好。但是，如果要输出 pdf 文档或 word 文档，并且公式、图表的编号还要交互引用的话，就可能有问题。所以，一般情况下，建议用 LaTeX 的公式环境。

### 6.2.2 LaTeX 公式环境

一行整行公式：公式自动居中。

```
\begin{equation}
y=f(x)
\end{equation}
```

$$y = f(x) \quad (1)$$

关于公式编号：输出为 html 时，公式默认不编号，只有标注了的公式才编号。而输出为 pdf 时，一般默认为自动编号。

如果想输出的公式不编号，可以用 `equation*` 环境或在该公式后面增加 `\nonumber` 或 `\notag`。其它公式环境类似，如 `eqnarray*`, `gather*`, `align*` 等。

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

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

多行公式：用 `eqnarray`，等号对齐用 `&=`，换行用 `\\`。

$$y = f(x) + c \quad (2)$$

$$z = g(y) + \sin(x) \quad (3)$$

公式对齐可以用 `align`，等号对齐用 `&=`；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) \notag \\
\sqrt{n}[g(X_{\{n\}}) - g(\theta)] &= g'(\tilde{\theta}) \sqrt{n}[X_{\{n\}} - \theta] \\
&\quad (\text{\#eq:grad})
\end{align}
```

$$\begin{aligned} g(X_n) &= g(\theta) + g'(\tilde{\theta})(X_n - \theta) \\ \sqrt{n}[g(X_n) - g(\theta)] &= g'(\tilde{\theta}) \sqrt{n}[X_n - \theta] \end{aligned} \quad (4)$$

如果希望多行公式共享一个编号，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'X)^{-1} \\
&= (X'X)^{-1} \sigma^2
\end{split}
\end{equation}
```

$$\begin{aligned}
 \text{Var}(\hat{\beta}) &= \text{Var}((X'X)^{-1}X'y) \\
 &= (X'X)^{-1}X'\text{Var}(y)((X'X)^{-1}X')' \\
 &= (X'X)^{-1}\sigma^2
 \end{aligned}
 \tag{5}$$

## 6.2.3 参考手册

LaTeX 参考手册: <https://wch.github.io/latexsheet/latexsheet-a4.pdf>

RStudio 的 Help 还有很多使用手册 (Cheatsheets), 也可以去下载: <https://www.rstudio.com/resources/cheatsheets/>。

## 6.3 R 代码块

行内 R 代码: ``r R_CODE``。如 ``r exp(sin(pi/2))``, 计算结果为  $e^{\sin(\pi/2)} = 2.7183$ 。

插入 R 代码块: 在 RStudio 左上角的 **Insert** 可插入 R 代码 (也可以是 Python、Stan 等), 或者用快捷键 **Ctrl + Alt + I**。

代码块可以运行代码并输出文本结果、图形和表格, 使得 R 和 Markdown 方便的放在一起<sup>5</sup>。

```
```${r}
```
```

代码块有丰富的控制选项, 可参考 *knitr* 说明文档 <https://yihui.name/knitr/options>。

代码块常用选项:

- **eval**: 代码块是否运行。**eval=FALSE** 代码仅显示而不实际运行。
- **echo**: 是否在结果中显示源代码。
- **results**: 用选项 **results=** 选择文本型结果的类型。取值有:
  - **markup**, 这是缺省选项, 会把文本型结果变成 HTML 的原样文本格式。
  - **hide**, 运行了代码后不显示运行结果。
  - **hold**, 一个代码块所有的代码都显示完, 才显示所有的结果。
  - **asis**, “as-is”, 直接输出源代码文本。*knitr* 包的 **kable()** 函数可以把数据框转换为 HTML 表格。
- **collapse**: 是否把源代码和输出文本放在一个输出文本块。默认是分开的。
- **warning, message, and error**: 是否显示警告、消息和错误信息。
- **include**: 代码块和结果是否包含在输出文档中。当 **include = FALSE**, 但 **eval = TRUE**, 则代码块和结果都不出现在输出文档里, 但仍然运行。
- **cache**: 是否缓存。部分修改时, 仅运行修改部分。
- **fig.width**: R 图形本身的宽度 (英寸), 如 **fig.width = 6**。
- **fig.height**: R 图形的高度, 如 **fig.height = 4**。
- **fig.asp**: 表示图形的高宽比。
- **out.width**: R 图形在生成文档中的宽度。如 **out.width = '80%'** 表示占页面宽度的 80%。
- **out.height**: R 图形在生成文档中的高度。
- **fig.align**: 图形的对齐方式, 可以是 **'left'**, **'center'**, or **'right'**。
- **dev**: R 图形的装置。常见的有 **'pdf'** (for LaTeX output)、**'png'** (for HTML output)、**'svg'** (矢量图格式)、**'jpeg'** (压缩格式), 本文用 **'cairo\_pdf'**, 可以在图形中出现中文。
- **fig.cap**: 图形和表格的标题。
- **child**: 主文件中包含的子文件。

例如:

只显示代码: `{r eval=FALSE}`

显示代码 + 编译结果: `{r tidy=TRUE,warning=FALSE,message=FALSE}`

只要结果, 不显示代码、错误提示和报警: `{r echo=FALSE,warning=FALSE,message=FALSE}`

<sup>5</sup>为了显示在 R Markdown 中实际输入的代码块 (包括首尾标志), 需要将代码段整体地缩进 4 个空格, 并在三个反单撇号前面加上 ``r '``, 即生成一个空字符串的行内 R 代码。

输出时不显示代码和编译的结果，仅自己在 **Rmarkdown** 中调出结果或者用于预览：  
`{r include=FALSE}`

## 6.4 图形

### 6.4.1 插入图片

方式一：Markdown 插入。

方法简单，但不能自动编号，而且输出 html 和 pdf 时显示结果会不一致。

- 插入网络图片：
- 插入本地图片：（子目录 images 下），显示为：



方式二：R Markdown 插入。

用 `knitr` 函数 `knitr::include_graphics()` 插入图形（图2）（设置了宽度等于页面的 50%，居中）。

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

用 `include_graphics()` 插入图形有如下优点：

1. 不必担心文档的输出格式。LaTeX 的 pdf 中插入图形命令是 `\includegraphics{}`，在 Markdown 中插入图形命令是 `![]()`，它们互相不认。而函数 `include_graphics()` 会自动选择插入方式。
2. 无论是插入图形还是由 R 生成的图形，控制图形属性的指令（`out.width` 等）都含义相同。
3. 可以对所有图形设置相同的大小比例。如本文设置 `out.width = '75%'`。

所以，最好是统一使用 Rmd 的插图方法。

### 6.4.2 R 做图

R 作图可以指定大小和对齐方式（见图3）。



图 2: 插入本地图形

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

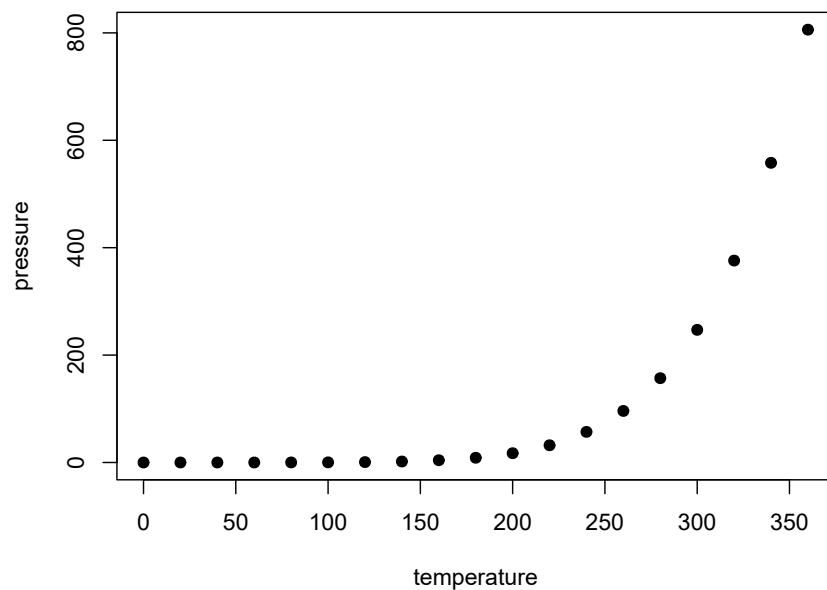


图 3: 图形指定大小和位置

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%<sup>6</sup>。

见图4。

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

<sup>6</sup>因为 word 不允许多子图，所以本文无法输出 word 格式。如果 word 确实需要多个子图，可以先保存到本地，然后再插入。



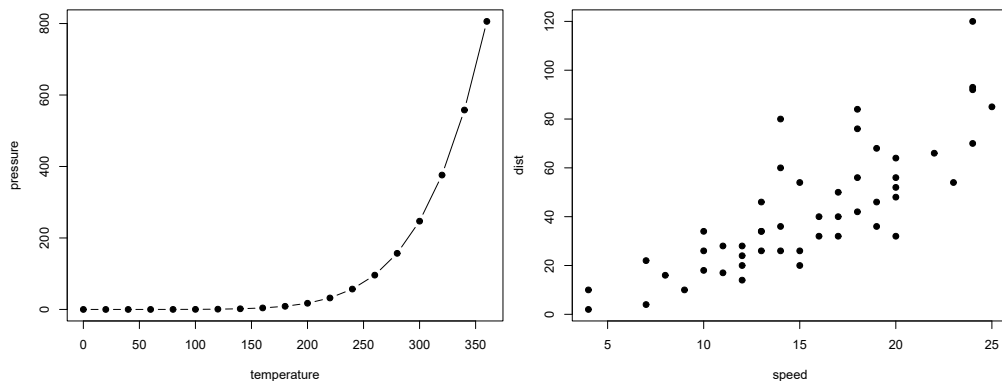


图 4: 两个图形并排

### 6.4.3 一个 ggplot2 作图的例子

软件包 `ggplot2` (Wickham et al. 2018) 是 R 中可视化的利器，作者是 Hadley Wickham。“gg”表示“Grammar of Graphics”的意思，作图理念是分图层，

Plot (图形) = data (数据) + Aesthetics (美学) + Geometry (几何)

包括:

- data 指数据框 (data frame)
- Aesthetics 标明 x 和 y 变量，以及控制图形的颜色、大小、数据点的形状、柱形的高低等美学特征
- Geometry 表示图形的几何形状 (直方图、箱线图、密度图等)
- 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

下面具体介绍一个用 `ggplot2` 作图的例子<sup>7</sup>。

#### 6.4.3.1 散点图

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

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5)。

`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

#### 6.4.3.2 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 6。

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

<sup>7</sup>这个例子来自 [https://blog.csdn.net/weixin\\_41929524/article/details/79765882](https://blog.csdn.net/weixin_41929524/article/details/79765882)。

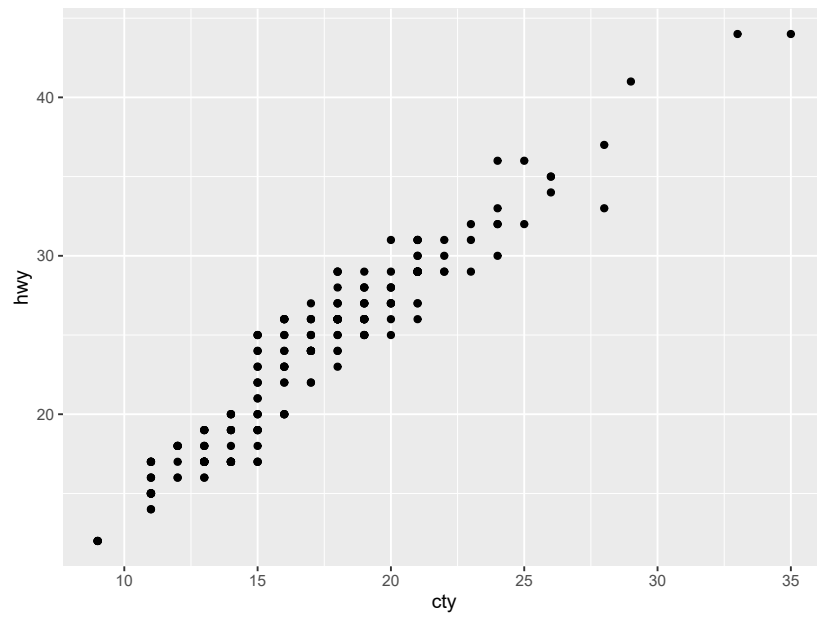


图 5: ggplot2 with points

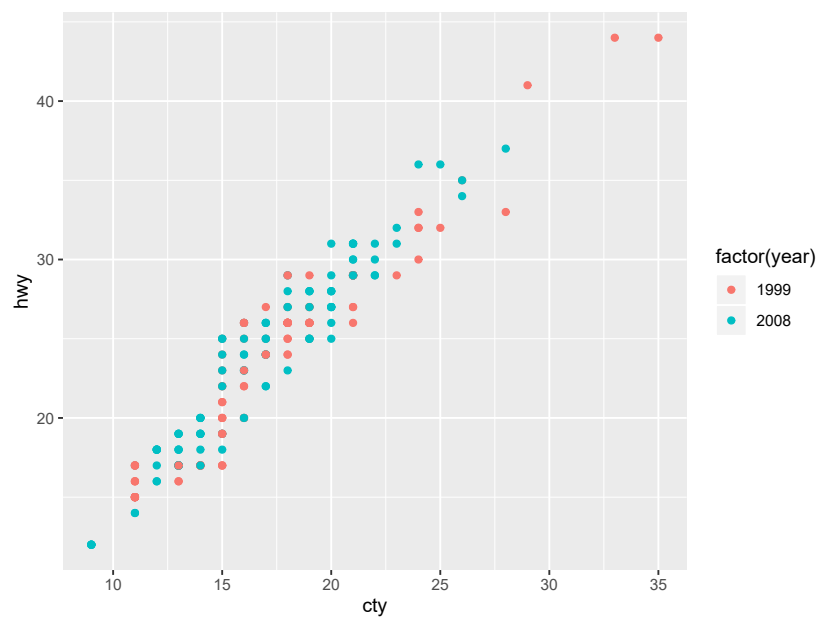


图 6: ggplot2 with points colors

### 6.4.3.3 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 7。

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

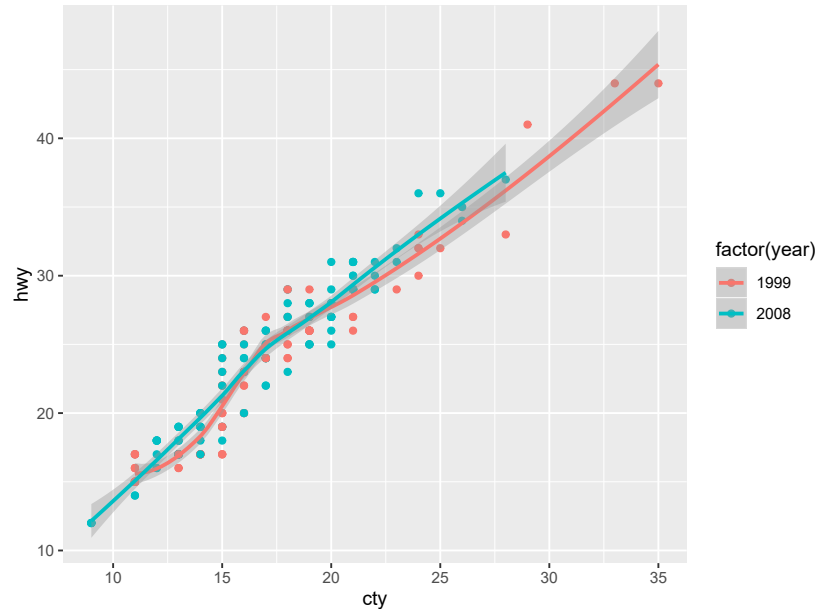


图 7: ggplot2 with smooth trends

### 6.4.3.4 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 8。

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

### 6.4.3.5 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 9。

```
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。

### 6.4.3.6 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 10。

```
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。

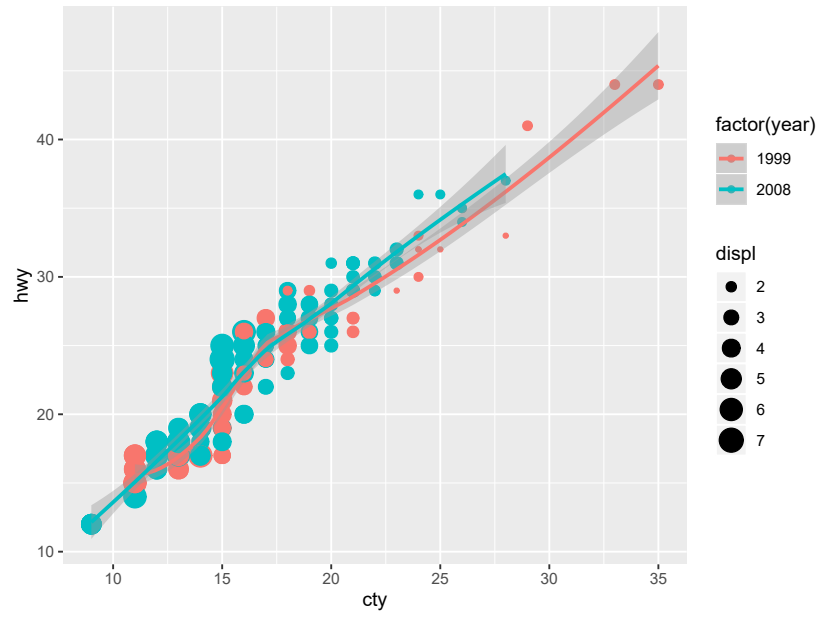


图 8: ggplot2 with variable point size

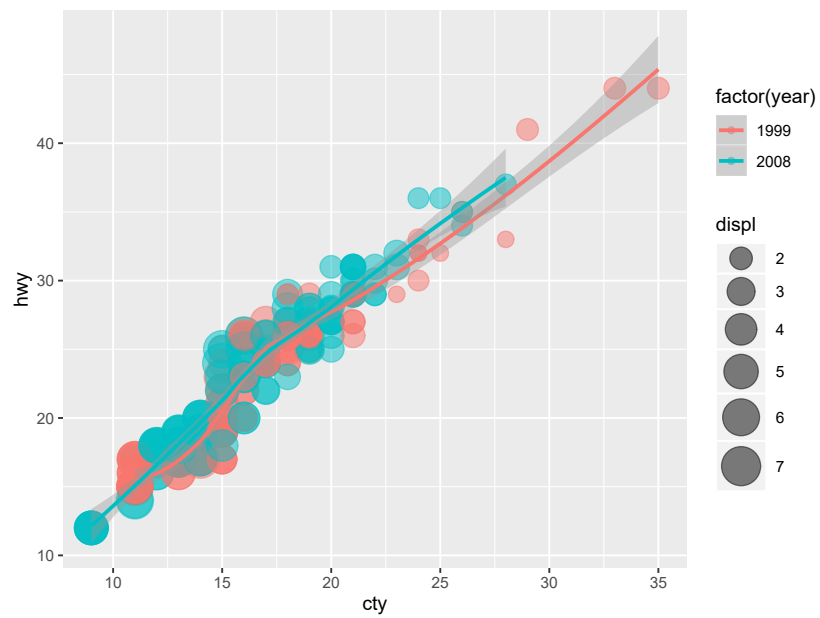


图 9: ggplot2 with transparent points

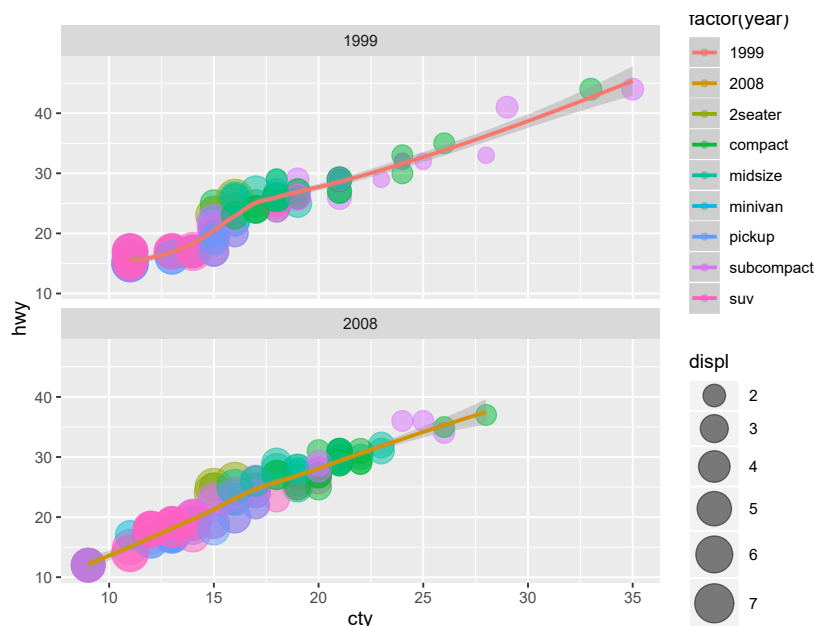


图 10: ggplot2 with facets

- 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。

#### 6.4.3.7 添加中文标注

默认情况下图形是不能出现中文的, 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 R 代码中加上 `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 11 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。

## 6.5 表格

### 6.5.1 用 R Markdown 函数 `kable()` 制作表格

用法: `knitr::kable()`

这是 R Markdown 显示表格最方便的方式, 可以把数据框或模型结果转化成有格式的表格, 支持 HTML、docx、LaTeX 等格式, 并且表格有标题选项, 可以自动编号和交互引用, 例如表 1。

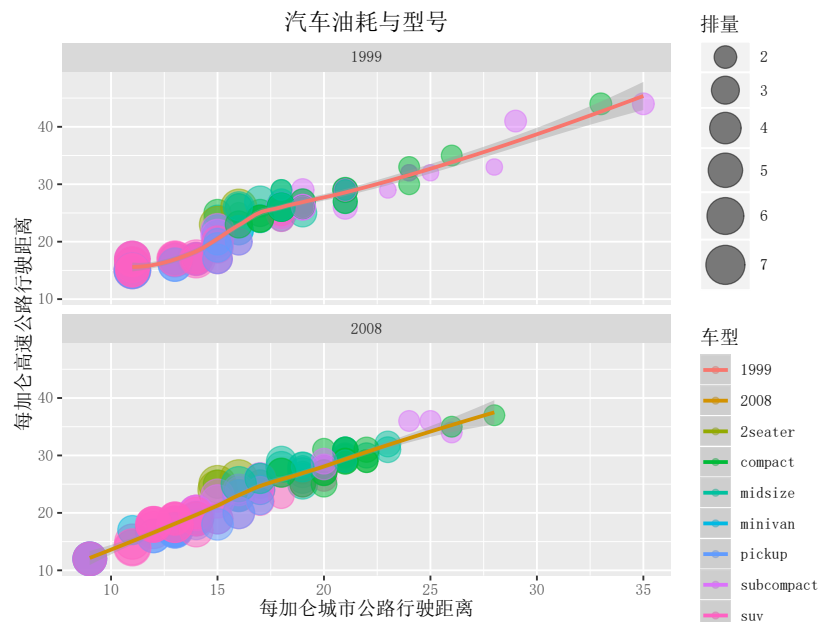


图 11: ggplot2 中显示中文

表 1: mtcars 数据的前 10 行

|                   | 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  |

```
knitr::kable(
 head(mtcars[, 1:8], 10), booktabs = TRUE,
 caption = 'mtcars 数据的前 10 行'
)
```

回归模型的回归系数用表格显示 (表2):

```
x <- rnorm(100, mean=100, sd=10)
y <- 2*x + 3*rnorm(100)
out <- lm(y~x)
cofs <- summary(out)$coefficients
knitr::kable(cofs, booktabs = TRUE, caption = ' 回归系数及显著性检验结果')
```

在 LaTeX 的 pdf 中, 表格和图形经常会浮动位置。如果不希望表格浮动 (因此表格允许跨页), 可以用 LaTeX 软件包 `longtable`。当然, 在 html 网页中不会出现这种情况, 因为网页是不分页的。

用法: 在 `kable()` 中指定 `longtable = TRUE`。如

```
```{r longtable, tidy=FALSE}
knitr::kable(
```

表 2: 回归系数及显著性检验结果

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.5689	3.0663	0.1855	0.8532
x	1.9981	0.0304	65.6313	0.0000

```
iris[1:10, ], longtable = TRUE, booktabs = TRUE,
caption = '用longtable包生成的一个表格'
)
...
```

6.5.2 用 R 软件包 xtable 和 stargazer 制作表格

R 扩展包 `xtable` 提供了一个 `xtable()` 函数, 可以用来生成 HTML 格式和 LaTeX 格式的表格, 但是需要指定输出格式 (latex, html 或 text), 不像 `kable` 可以同时输出多种格式。如果输出 pdf, 需在代码块选项中选择 `results="asis"`, 并且预先加载软件包 `library(xtable)`。

`xtable` 不仅对数据框, 还可以对许多统计模型的结果输出表格, 包括回归模型、方差分析、主成分分析等。

还有一个软件包 `stargazer` 与 `xtable` 类似。

6.5.3 Markdown 表格

Markdown 表格由简单文本表示, 转化为 HTML、docx 等格式后就变成了富文本的表格。Markdown 表格默认是不编号的, 不能交互引用。为了使得其可交互引用, 需要在表标题的 `Table:` 处加上引用标志, 如 `(\#tab:label)`。

6.5.3.1 管道表

管道表在两列之间用竖线分开, 在列标题下面用减号画横线, 用如下方法指定各对齐方式:

- 在列标题下的横线开始加冒号, 表示左对齐;
- 在列标题下的横线末尾加冒号, 表示右对齐;
- 在列标题下的横线两端加冒号, 表示居中对齐;
- 列标题下面仅有横线没有冒号, 表示缺省对齐方式, 一般是左对齐。

这种方法不需要输入内容上下对齐, 适用于中文内容 (表3)。

表 3: 管道表

姓名	收入	职业	颜色偏好
赵四海	123456	业务经理	红
刘英	50	无	蓝
钱德里	3200	保洁	灰

6.5.3.2 简单表

简单表的格式是, 第一行是各列标题, 第二行是各标题下面用减号组成的表格线, 同一行的不同列要用空格分开, 从第三行开始是内容。

在表格前或表格后用空行隔开的以 `Table:` 开头的行是表格说明或标题。

为了确定表格每列单元格内容如何对齐, 用列标题下的表格线给出提示:

- 表格线与列标题右对齐, 表示该列右对齐;
- 表格线与列标题左对齐, 表示该列左对齐;
- 列标题在表格线中间, 表示该列居中对齐;
- 列标题左右都与表格线对齐, 表示该列为缺省对齐方式, 一般是左对齐。
- 一定要使用一个等宽字体来编辑这样的表格, 否则对齐与否无法准确分辨。
- 单元格内容不能超出表格线左端。

经过试验发现, 英文没问题, 中文内容很难按这种方法对齐 (表4)。

表 4: 一个简单 Markdown 表格

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

6.5.3.3 换行表

换行表在输入列标题和单元格内容时，允许输入内容拆分行，但是编译后并不拆分行。这样的表以一行减号开始，以一行减号结束，中间的表格用空行分开实际的不同行。例如（表5）：

表 5: 一个换行表的例子

Name of Subject	Income	Job	color
Jane Ayer	123456	Research Assistant	red
John Tukey	50	N/A	blue
William Tale	3200	Cleaner	blue

换行表输入时各列的输入宽度是有作用的，输入较宽的列结果也较宽。

6.5.3.4 有格表

完全用减号、竖线、等于号、加号画出表格线。这样的表在文本格式下呈现出很好的表格形状。编译后不能指定对齐方式（表6）。

表 6: 一个有格表

水果	价格	优点
Bananas	\$1.34	<ul style="list-style-type: none"> • built-in wrapper • bright color
Oranges	\$2.10	<ul style="list-style-type: none"> • cures scurvy • tasty

6.6 交叉引用

R Markdown 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、表格和章节的标记和引用。

6.6.1 数学公式

要给公式编号，先要建立公式环境。如果公式环境是 $...$ ，那么输出 html 完全没问题，但是 pdf 和 word 都表现不佳。所以，如果要输出 pdf 格式，最好统一建立 LaTeX 公式环境，即 $\begin{equation}...\end{equation}$ 等。

然后标记 ($\#eq:label$)，例如

```
\begin{equation}
f\left(k\right) = \binom{n}{k} p^k\left(1-p\right)^{n-k}
```



```
(\#eq:binom)
\end{equation}
```

结果如下:

$$f(k) = \binom{n}{k} p^k (1-p)^{n-k} \quad (6)$$

引用方法是`\@ref{eq:binom}`，如，参看公式 (6)。

6.6.2 定理

用下面形式创建一个定理环境:

```
\begin{theorem, label="mylabel", name=" 定理名"}
这是黎曼猜想。
\end{theorem}
```

一个定理的例子:

```
\begin{theorem, label="pyth", name=" 勾股定理"}
对于直角三角形，设  $c$  表示斜边的长度， $a$  和  $b$  分别表示两条直角边的长度，则
 $a^2 + b^2 = c^2$ 
\end{theorem}
```

结果如下:

定理 6.1 (勾股定理) 对于直角三角形，设 c 表示斜边的长度， a 和 b 分别表示两条直角边的长度，则有

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

引用用法：在需要引用处插入`\@ref{thm:label}`。比如，定理6.1是个著名的定理。

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

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

特征函数是个极为重要的函数，见定义1。

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

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

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.,

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

6.6.3 交互引用方式列表

对不同类型环境的交叉引用，大同小异，唯有对章节的引用区别较大。归纳为下表。

表 7: 交叉引用的方法列表

类型	标记	引用
公式	<code>(\#eq:label)</code>	<code>\@ref(eq:label)</code>
图形	<code>(\#fig:label)</code>	<code>\@ref(fig:label)</code>
表格	<code>(\#tab:label)</code>	<code>\@ref(tab:label)</code>
定理	<code>(\#thm:label)</code>	<code>\@ref(thm:label)</code>
章节	<code>{#label}</code>	<code>\@ref(label)</code>

6.7 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。

参考文献在正文中的插入非常简单，只要要在要引用的地方使用 `[@reference key]`，就能实现引用。如果同一个地方要引用多个文献，则是 `[@reference key1; @reference key2]`。在 `bib` 文档中，如 `@Manual{R-ggplot2}`，其中的 `R-ggplot2` 就是 `reference key`，复制粘贴到 `[@]` 里面即可。

引用方式：

一般引用有两种方式，一种是作者-年格式，另一种是数字格式。当然，一篇文章内要统一用一种格式。

作者-年格式 (apalike 样式): `@key` 表示作者姓名 (年份)，而 `[@key]` 则表示 (作者年份)。

数字编号格式 (chinese-gb7714-1987-numeric.csl 样式): `@key` 表示 [序号]，`[@key]` 则表示 [序号] 为上标。

其它格式: `@key` 表示 [序号]，`[@key]` 表示上标 [序号]，网页 [序号] 上标 `^[@key]`。

示例 (chinese-gb7714-1987-numeric.csl 样式): `@R-rmarkdown` 结果显示 Allaire et al. (2018); `[@R-rmarkdown;@xie2015;@R-ggplot2]` 结果显示 (Allaire et al. 2018; Xie 2015; Wickham 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。
- 一些常用软件包也可以用 `knitr` 的函数 `write_bib()` 产生。下面代码产生 `knitr`、`bookdown`、`tidyverse` 和 `ggplot2` 的 bibtex 格式文献。

```
# the second argument can be a .bib file
knitr::write_bib(c('knitr', 'bookdown', 'tidyverse', 'ggplot2'),
                 'bib/mypackages.bib', width = 60)
```

文献引用要在 R Markdown 文档的开头的 `yaml` 部分指定文献文档名 (`.bib` 文件) 和引用风格 (`biblio-style`)。

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

7 创建 Git 版本跟踪项目

Rstudio 已经聚合了 Git 跟踪项目，可以方便地进行版本控制。

首先下载并安装 Git (<https://git-scm.com/download/>)。

登进 Github. 新建一个仓库（或打开旧仓库），点开 Clone or Download, 右边拷贝 SSH clone URL(注意不是 https url!)（见图12）。

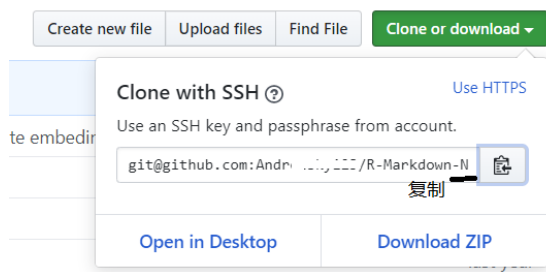


图 12: 拷贝 SSH 地址到 RStudio

打开 RStudio, File -> New Project（如果已经存在与 Github 仓库相同的 Project, 直接打开 .Proj 文件）, 打开-> Version Control -> Git -> Paste the SSH clone URL。注意 New Project 的名字和 Github 仓库名字要一致。

创建成功后, Rstudio 自动将工作目录转到新建的项目, 并且在右上操作框出现 Git 按键, 通过 Rstudio 或者 Github Desktop 来实现 Github 托管项目都可以（见图13）。

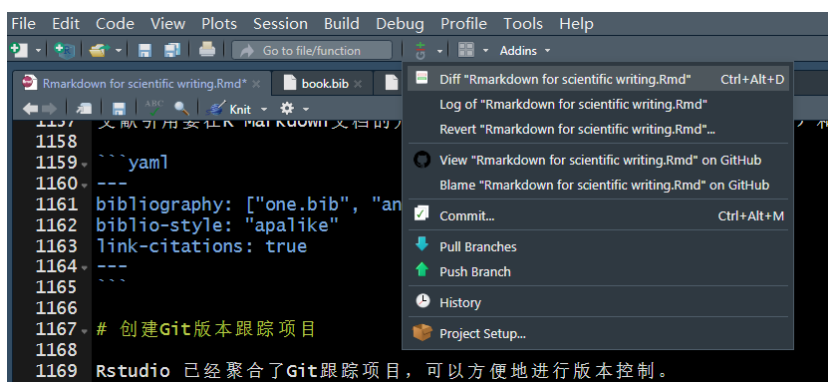


图 13: RStudio 的 git 操作界面

8 总结

本文简单介绍了如何用 R Markdown 写一般性研究短文（标题、公式、图表、参考文献等都能自动排版和编号）、中等篇幅的研究论文（可以交互引用）和长篇幅的研究文章（书籍、研究报告、学位论文等）等。还介绍了 R Markdown 的一些基本的用法，以方便在写作过程中随时参考。

用 R Markdown 写研究性文章最大的优点是可以把数据分析的过程（不局限于 R，也可以是 Python 等）和文章文本结合起来，写作过程中可以方便地进行修改，保持文章结果的可重复性。

LaTeX 的一些前端（如 LyX）也可以做到 R 运行与文本的结合，但必须全文编译后才能看到结果，不如 R Markdown 方便，可以在中间过程运行 R 代码块，随时查看部分计算结果。

总之，R Markdown 简单易学，是做学习笔记、科学研究、撰写论文和书籍的好工具，值得花时间掌握。

参考文献

- Allaire, JJ, Yihui Xie, Jonathan McPherson, Javier Luraschi, Kevin Ushey, Aron Atkins, Hadley Wickham, Joe Cheng, and Winston Chang. 2018. *Rmarkdown: Dynamic Documents for R*. <https://CRAN.R-project.org/package=rmarkdown>.
- R Core Team. 2018. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. <https://www.R-project.org/>.
- Wickham, Hadley, Winston Chang, Lionel Henry, Thomas Lin Pedersen, Kohske Takahashi, Claus Wilke, and Kara Woo. 2018. *Ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics*. <https://CRAN.R-project.org/package=ggplot2>.
- Xie, Yihui. 2015. *Dynamic Documents with R and Knitr*. 2nd ed. Boca Raton, Florida: Chapman; Hall/CRC. <http://yihui.name/knitr/>.
- . 2018a. *Bookdown: Authoring Books and Technical Documents with R Markdown*. <https://CRAN.R-project.org/package=bookdown>.
- . 2018b. *Knitr: A General-Purpose Package for Dynamic Report Generation in R*. <https://CRAN.R-project.org/package=knitr>.
- Zhao, Peng. 2017. *Bookdownplus: Generate Varied Books and Documents with R 'Bookdown' Package*. <https://CRAN.R-project.org/package=bookdownplus>.
- 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): 146–60.