R Markdown 简介

WSJ

College of Economics，Shenzhen University

Table of Contents

[1 基本语法 2](#_Toc524943178)

[1.1 行内语法 2](#_Toc524943179)

[1.2 模块语法 2](#_Toc524943180)

[2 数学公式 3](#_Toc524943181)

[3 R代码块 4](#_Toc524943182)

[4 图形 6](#_Toc524943183)

[4.1 R 代码块做图 6](#_Toc524943184)

[4.2 插入图形 8](#_Toc524943185)

[5 交叉引用 10](#_Toc524943186)

[5.1 数学公式 10](#_Toc524943187)

[5.2 定理 11](#_Toc524943188)

[6 一个ggplot2作图的例子 13](#_Toc524943189)

[6.1 散点图 13](#_Toc524943190)

[6.2 变换颜色 14](#_Toc524943191)

[6.3 图形分层 18](#_Toc524943192)

[6.4 添加中文标注 19](#_Toc524943193)

[7 表格 20](#_Toc524943194)

[7.1 用R函数kable()制作表格 20](#_Toc524943195)

[7.2 制作Markdown表格 21](#_Toc524943196)

[8 文献引用 22](#_Toc524943197)

[9 参考文献 25](#_Toc524943198)

# 基本语法

## 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 H2SO4
* 上标：两个^包夹， R^2^ renders R2
* 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](https://www.rstudio.com)
* 插入图片：![image title](path/to/image)
* 脚注：^[], e.g., 这里是脚注[[1]](#footnote-1)

## 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  
 - 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

# 数学公式

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

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

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

$$X = \begin{bmatrix}1 & x\_{1}\\  
1 & x\_{2}\\  
1 & x\_{3}  
\end{bmatrix}$$

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

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

# 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.

# 图形

## 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.

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

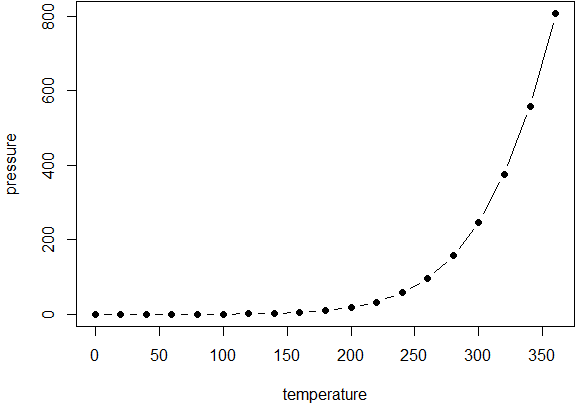


Figure 1 A figure example with the specified aspect ratio, width, and alignment.

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%.

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

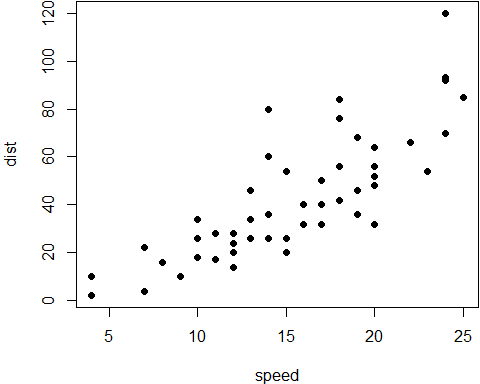


Figure 2 A figure example with a relative width 70%.

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% （word似乎不行）。

## 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。

knitr::include\_graphics('images/run.jpg')



Figure 3 image included in the document from an external PNG file.

用 include\_graphics() 插入图形有几大优点：

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).

# 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)。

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

结果如下:

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\*}

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（但word例外）

\begin{align}   
g(X\_{n}) &= g(\theta)+g'({\tilde{\theta}})(X\_{n}-\theta) \\  
\sqrt{n}[g(X\_{n})-g(\theta)] &= g'\left({\tilde{\theta}}\right)  
 \sqrt{n}[X\_{n}-\theta ] (\#eq:align)  
\end{align}

结果如下：

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^{2}IX(X'X)^{-1}\\  
 & =(X'X)^{-1}\sigma^{2}  
\end{split}  
(\#eq:var-beta)  
\end{equation}

## 定理

用下面形式创建一个定理环境:

```{theorem，label="yourlabel",name="theorem name"}  
Here is my theorem.  
```

一个定理的例子：

Theorem 1 (Pythagorean theorem) For a right triangle, if denotes the length of the hypotenuse and and denote the lengths of the other two sides, we have

引用：在需引用处插入 \@ref(thm:label)。

参看定理 1，见定义1。

定义、推论、命题、公里、假设等类似。

注意，如果 echo设置为FALSE，定理编号失效。

Definition 1 随机变量 的特征函数定义为

Example 1 We derive the characteristic function of with the probability density function .

Note that we used the fact twice.

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

Theorem 2 If , …, are independent random variables, and , …, are some constants, then the characteristic function of the linear combination is

Proposition 1 The distribution of the sum of independent Poisson random variables is .

This is the characteristic function of a Poisson random variable with the parameter . From Lemma 1, we know the distribution of is .

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

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

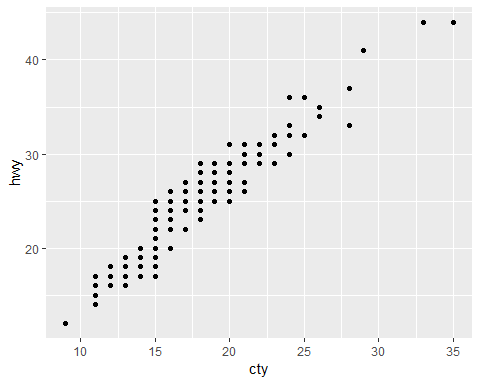
# 一个ggplot2作图的例子

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

## 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4）。

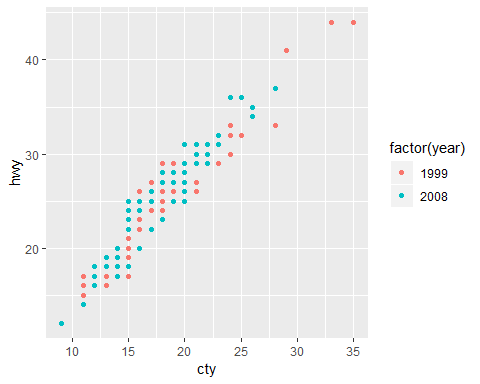
mpg是ggplot2里面的一个关于汽车的数据集。

* 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

## 变换颜色

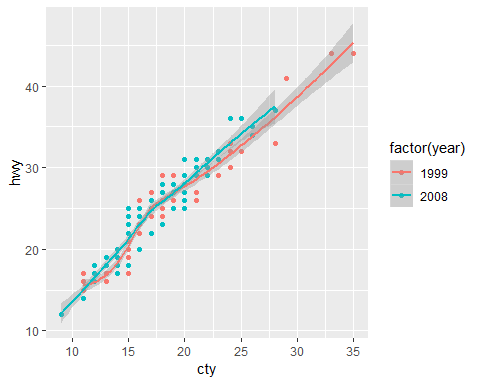
按生产年份以颜色区分，factor(year))是把年份转化为因子形式（相当于定类变量）, 见图 5。

p <- ggplot(mpg, aes(x = cty, y = hwy, colour = factor(year)))  
p + geom\_point()

 ## 拟合曲线

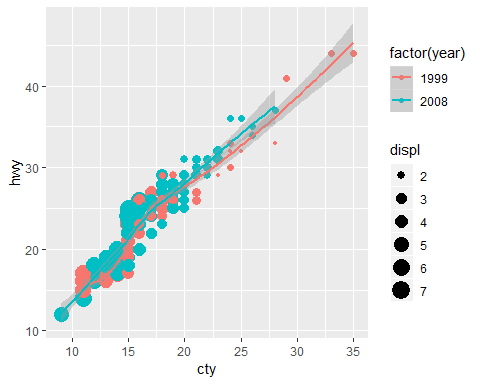
再加一行 + stat\_smooth()，其中stat表示 statistical transformation，做了统计平滑拟合直线，以及置信区间, 见图 6。

p + geom\_point() + stat\_smooth()

 ## 变换大小

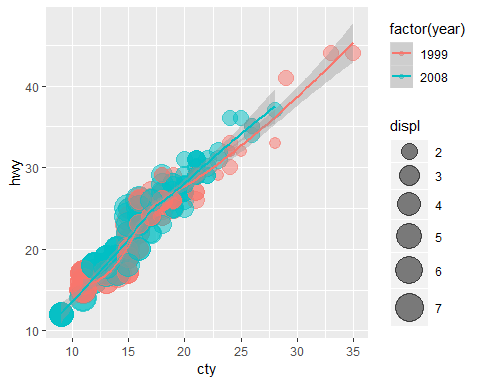
上图的数据点明显偏小，可以让这些数据点随着**汽车排量**的大小而变化, 见图 7。

p + geom\_point(aes(colour = factor(year), size = displ)) +   
 stat\_smooth() # 排量越大，点越大

 ## 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 8。

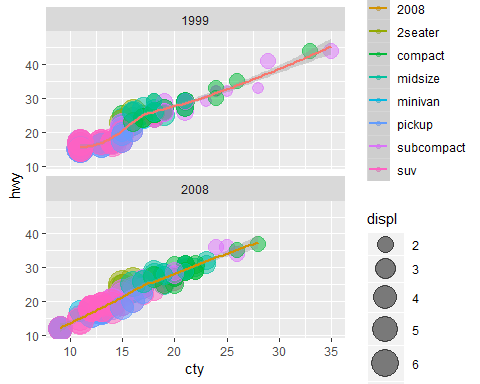
p + geom\_point(aes(colour = factor(year),  
 size = displ), alpha = 0.5) +  
 stat\_smooth() + scale\_size\_continuous(range = c(4, 10))

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

## 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 9。

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。

## 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))

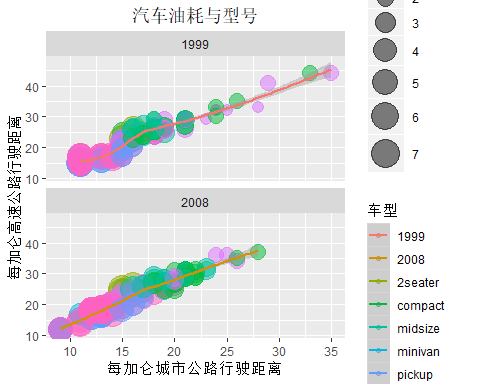


Figure 10 ggplot2 with Chinese characters

图 10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。

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

# 表格

## 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.'  
)

Table 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 |

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**,](https://www.ctan.org/pkg/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.'  
)

Table 2 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 |

## 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,](http://pandoc.org/MANUAL.html#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

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.

# 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 [[1](#ref-R-rmarkdown)]；示例[[1](#ref-R-rmarkdown)]。

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]](#footnote-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 [[2](#ref-R-base)], and [@R-base] generates “[[2](#ref-R-base)]”. 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 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）[[3](#ref-ke2017)]。

# 参考文献

[1] Allaire J, Xie Y, McPherson J, Luraschi J, Ushey K, Atkins A, Wickham H, Cheng J, Chang W. rmarkdown: Dynamic Documents for R[M]. 2018.

[2] R Core Team. R: A Language and Environment for Statistical Computing[M]. Vienna, Austria: R Foundation for Statistical Computing, 2018.

[3] 柯忠义. 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[J]. 数量经济技术经济研究, 2017, 34(1): 146–160.

1. This is a footnote. [↑](#footnote-ref-1)
2. The type name is case-insensitive, so it does not matter if it is manual, Manual, or MANUAL. [↑](#footnote-ref-2)