

華東師範大學

EAST CHINA NORMAL UNIVERSITY

博士学位论文 (统计学专业)

---

# 华东师范大学毕业论文 Quarto 模板

---

开发者:  
汤银才

更新时间:  
2024 年 6 月





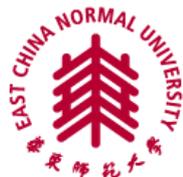
2024届研究生博士学位论文

分类号: O212.1

学校代码: 10269

密级:

学号: YS012345678



華東師範大學

EAST CHINA NORMAL UNIVERSITY

博士学位论文

研究生毕业论文题目

院 系: 统计学院

专 业: 统计学

研究 方 向: 应用统计

指 导 老 师: 某某某 教授

学 位 申 请 人: 李 某

2024 年 5 月





DOCTORAL DISSERTATION

Title for Graduate Dissertation

Department: 统计学院

Major: Statistics

Research Direction: Applied Statistics

Supervisor: Prof Wang xxx

Candidate: Li xx

May 2024

## 华东师范大学学位论文原创性声明

郑重声明：本人呈交的学位论文《研究生毕业论文题目》，是在华东师范大学攻读硕士/博士（请勾选）学位期间，在导师的指导下进行的研究工作及取得的研究成果。除文中已经注明引用的内容外，本论文不包含其他个人已经发表或撰写过的研究成果。对本文的研究做出重要贡献的个人和集体，均已在文中作了明确说明并表示谢意。

作者签名：\_\_\_\_\_ 日期：\_\_\_\_年\_\_月\_\_日

## 华东师范大学学位论文著作权使用声明

《研究生毕业论文题目》系本人在华东师范大学攻读学位期间在导师指导下完成的硕士/博士（请勾选）学位论文，本论文的著作权归本人所有。本人同意华东师范大学根据相关规定保留和使用此学位论文，并向主管部门和学校指定的相关机构送交学位论文的印刷版和电子版；允许学位论文进入华东师范大学图书馆及数据库被查阅、借阅；同意学校将学位论文加入全国博士、硕士学位论文共建单位数据库进行检索，将学位论文的标题和摘要汇编出版，采用影印、缩印或者其它方式合理复制学位论文。

本学位论文属于（请勾选）

- ( ) 1. 经华东师范大学相关部门审查核定的“内部”或“涉密”学位论文\*，  
于\_\_\_\_年\_\_月\_\_日解密，解密后适用上述授权。
- ( ) 2. 不保密，适用上述授权。

导师签名：\_\_\_\_\_ 本人签名：\_\_\_\_\_

\_\_\_\_年\_\_月\_\_日

\*“涉密”学位论文应是已经华东师范大学学位评定委员会办公室或保密委员会审定过的学位论文（需附获批的《华东师范大学研究生申请学位论文“涉密”审批表》方为有效），未经上述部门审定的学位论文均为公开学位论文。此声明栏不填写的，默认为公开学位论文，均适用上述授权）。

李 某 博士学位论文答辩委员会成员名单

姓 名	职 称	单 位	备 注
			主 席



# 目录

摘 要	i
ABSTRACT (英文摘要)	iii
<b>第一章 第一章标题</b>	<b>1</b>
1.1 欢迎和感谢	1
1.2 LaTeX 简明数学指南	1
1.3 关于这个模板	1
1.4 这个模板包括什么	2
1.5 填写你的信息	2
1.6 <code>tex\before-body.tex</code> 文件解释	3
1.7 添加您的章节和附录	3
1.8 参考文献和引用	4
1.9 论文特性和约定	4
1.10 表格	5
1.11 图表	6
1.12 结束语	6
<b>第二章 Quarto 书稿速览</b>	<b>7</b>
2.1 Quarto 的结构	7
2.2 关于 Quarto Book	8
2.3 书的基本设置	8
2.4 章节结构	13
2.5 书的编译	14
2.6 浮动对象标签与引用汇总	15
<b>第三章 Quarto 书稿中的章节标题</b>	<b>17</b>
3.1 章节标题	17
3.2 章节标题标签的设定与引用	17
<b>第四章 Quarto 书稿中的公式与定理</b>	<b>19</b>
4.1 公式标签的设定与引用	19
4.2 定理标签的设定与引用	22
4.3 数学公式的扩展	22
<b>第五章 Quarto 书稿中的图形的处理</b>	<b>23</b>
5.1 由 R 生成单个图形示例	23
5.2 由 R 生成两个图形并置或堆叠	24
5.3 单个静态图形的插入	25
5.4 静态图形的并置	25
5.5 图形引用	28
<b>第六章 Quarto 书稿中的表格的处理</b>	<b>29</b>
6.1 表格示例 1: 由数据生成表格	29
6.2 表格示例 2: 由数据框构造表格	30
6.3 表格示例 3: 由 markdown 表格直接生成	30
6.4 markdown 子表的并置	31
6.5 数据子表的并置与堆叠	32

6.6 表格引用 .....	33
<b>附录</b>	<b>35</b>
<b>附录 A 附录标题</b>	<b>35</b>
A.1 附录中的图形、表格、公式 .....	35
A.2 R 代码 .....	35
A.3 Python 代码 .....	35
<b>附录 B Mathjax 的离线安装与使用</b>	<b>37</b>
B.1 mathjax 简介 .....	37
B.2 调用远程服务器上的 mathjax .....	38
B.3 mathjax 本地服务器的安装与使用 .....	38
<b>附录 C 基于 biblatex 生成参考文献</b>	<b>41</b>
C.1 概述 .....	41
C.2 biblatex 的在 TeX 中的使用样例 .....	42
C.3 biblatex 在 Quarto 书稿中的使用 .....	43
<b>参考文献</b>	<b>45</b>
<b>在学期期间的研究成果及发表的论文</b>	<b>47</b>
在国际和国内学术刊物上发表的论文 .....	47
待发表论文 .....	47
<b>致谢</b>	<b>49</b>

# 插图目录

图 2.1 R Bookdown 编译界面.....	15
图 5.1 iris 数据集 <code>Petal.Length ~ Species</code> 的箱线图.....	23
图 5.2 iris 数据集 <code>Petal.Length ~ Species</code> 的散点图. 右侧的图像中散点类型通过 <code>Species</code> 因子的水平给出, 见图例. 直线为数据集拟合线性模型的结果.....	24
图 5.3 iris 数据集 <code>Petal.Length ~ Species</code> 的散点图. 下图的图像中散点类型通过 <code>Species</code> 因子的水平给出, 见图例. 直线为数据集拟合线性模型的结果.....	26
图 5.4 R logo.....	27
图 5.5 数据科学中的编程语言.....	27
图 5.6 数据科学中的编程语言.....	28



## 表格目录

表 1.1	这是我的标题。 .....	5
表 6.1	mtcars 数据的前 5 行 .....	29
表 6.2	遗传连接模型例子中观测到的频数 $y_i$ 和频率 $p(y_i \pi)$ , $i = 1, \dots, 4$ , 197 个 动物 .....	30
表 6.3	INLA 提供的一些似然 .....	31
表 6.4	二个子表的并置 .....	32
表 6.5	两个表格并置示例 2 .....	33
表 6.6	两个表格堆叠示例 .....	34



## 摘 要

这里是中文摘要，这里是中文摘要，这里是中文摘要，这里是中文摘要，这里是中文摘要，这里是中文摘要，这里是中文摘要。

请陈述你的研究内容与主要结论。

一般而言，中文摘要包含 500 - 1000 字，1 - 2 页。关键词 5 - 10 个。

**关键词：** 关键词 1, 关键词 2, 关键词 3, 关键词 4, 关键词 5



# Abstract

Here is the English abstract. Here is the English abstract. Here is the English abstract.  
Here is the English abstract. Here is the English abstract.

Please state your research contents and main results.

**Key Words:** Keyword 1, Keyword 2, Keyword 3, Keyword 4, Keyword 5



## 主要符号对照表

Ph.D. 哲学博士 (Doctor of Philosophy)  
M.S. 理学硕士 (Master of Science)



# 第一章 第一章标题

这是第 一 章的内容 (茆诗松 and 汤银才 2012; 韦来生 and 张伟平 2022)

## 1.1 欢迎和感谢

欢迎来到这个使用 LaTeX 排版系统和`Quarto`的 LaTeX 学位论文模板，基于从`LaTeXTemplates`下载的 LaTeX 学位论文模板 `MastersDoctoralThesis` 版本 2.0。这个 LaTeX 文档类由 Vel ([vel@latextemplates.com](mailto:vel@latextemplates.com)) 和 Johannes Böttcher 编写，基于英国南安普顿大学电子与计算机科学系的 Steve R. Gunn 的风格文件。

## 1.2 LaTeX 简明数学指南

如果你正在写一个技术或数学方面的论文，那么你可能想要阅读由 AMS (美国数学学会) 撰写的文档，“LaTeX 简明数学指南”。它可以在`AMS`网站上找到，位于页面底部的“附加文档”部分。

### 1.2.1 常见 LaTeX 数学符号

对于 LaTeX 来说，有大量的数学符号可用，学习所有这些符号的命令需要付出巨大的努力。你可能最常使用的一些符号显示在[this page](#)上。

你可以使用这个页面作为参考或速查表，符号以大号、高质量的图像显示，这样你可以快速找到所需的 LaTeX 符号命令。

## 1.3 关于这个模板

这个 LaTeX 学位论文模板最初是基于英国南安普顿大学电子与计算机科学系的 Steve R. Gunn 创建的 LaTeX 风格文件。你可以在这个网站找到他原始的学位论文风格文件：<http://www.ecs.soton.ac.uk/~srg/softwaretools/document/templates/>。

Steve 的 `ecsthesis.cls` 文件被 Sunil Patel 获取并修改，他创建了一个骨架框架和文件夹结构来放置学位论文文件。生成的模板可以在 Sunil 的网站上找到：<http://www.sunilpatel.co.uk/thesis-template>。

Sunil 的模板通过 [LaTeX Templates](#) 提供，并基于用户请求和问题进行了多次修改。版本 2.0 及以后的模板对 Sunil 的模板进行了重大修改，实际上已经难以辨认。使版本 2.0 成为可能的工作由 Vel ([vel@latextemplates.com](mailto:vel@latextemplates.com)) 和 Johannes Böttcher 完成。

## 1.4 这个模板包括什么

### 1.4.1 文件夹

- 附录 – 这个文件夹用于放置附录。每个附录应该放在单独的 qmd 文件中。目录中包含了一个示例和模板。
- 章节 – 这个文件夹用于放置学位论文的章节。每个章节应该放在单独的 qmd 文件中。
- 图表 – 这个文件夹包含学位论文的静态图表，即不是由章节中的代码生成的图表。

### 1.4.2 文件

- example.bib – 这个文件包含了你将在学位论文中引用的所有书目信息和参考文献，用于 BibTeX。你可以手动编写，但也有可用的参考文献管理程序会为你创建和管理。Zotero 是一个流行的选择，它可以与 RStudio IDE 集成。
- MastersDoctoralThesis.cls – 这个是告诉 LaTeX 如何格式化学位论文的类文件。
- pdf in docs folder – 这是你的排版学位论文。
- Frontmater 文件夹 – 这个文件夹包含了各种前言元素的文件。

## 1.5 填写你的信息

大部分个人信息可以在 `_quarto.yml` 文件中找到。

- author – 你；可选地添加网址
- supervisor – 你的指导教师；可选地添加网址
- university – 你的大学

- department – 你的系
- faculty – 学院名称
- group – 研究组名称（可选）

## 1.6 `tex\before-body.tex` 文件解释

`tex\before-body.tex` 文件包含了学位论文的结构，是 Pandoc 模板和 LaTeX 代码的混合体。看起来像 `$book.university$` 的部分是 Pandoc 的，它引用了 `_quarto.yml` 文件中的变量。知道了这一点，你应该能够弄清楚发生了什么。

文件中有许多注释解释了 LaTeX 代码创建了哪些页面、部分和格式。每个主要文档元素都被分为带有大写标题的注释块，以使接下来的代码块的作用显而易见。一开始可能看起来有很多 LaTeX 代码，但这都是格式设置，已经全部处理好了，所以你不需要做这些。

许多部分都有 `$if(...)$`，这样只有你在 `_quarto.yml` 中包含了该部分的信息，它才会被包含。

在 `_quarto.yml` 中，`pdf: toc: false` 被用来防止 Quarto/Pandoc 添加目录。这个模板将目录放在了缩略语和符号页之前，而 Quarto/Pandoc 不让我们控制目录的放置位置。因此，我们必须手动添加 PDF 的目录，并传入 `toc: false`。

列表中的图表和表格都已为您处理，不需要手动创建或编辑。接下来的几页更可能是可选的，并且可以删除，因为它们是为更技术性的论文准备的：插入您在论文中使用的缩略语列表，然后是您引用的物理常数和数字列表，最后是您在公式中使用的数学符号列表。努力填写这些表格意味着读者有一个一站式的参考地方，而不必在互联网和参考文献中搜索，试图找出您对某些缩略语或符号的意思。

符号列表分为罗马和希腊字母。而缩略语和符号应该按字母顺序列出（这不会自动为您完成），物理常数列表应该按相似主题分组。

下一页包含一行献词。您将把您的论文献给谁？

## 1.7 添加您的章节和附录

将您的章节和附录添加到 `_quarto.yml`。注意，间距很重要，前面必须有 `-`。

## 1.8 参考文献和引用

引用将自动为您添加和格式化。

如果您使用 RStudio IDE，那么您可以将 Zotero 与 RStudio 和 Quarto 链接起来，当您输入 @ 时，Quarto 将为您找到引用。这是在可视化编辑器模式中。确保搜索如何做到这一点的视频，因为使用 Zotero 库将使您的引用和参考文献管理变得更加容易。

在文本中使用 @smith2000 产生 Smith (2000)，使用 [@smith2000, @jones1999] 产生 (Smith 2000; Jones 1999)。有关如何在文本中引用的其他格式，请参阅 natbib 速查表。参考文献样式 (classoption: "authoryear") 用于参考文献，这是一个功能齐全的样式，甚至会包含链接到引用论文在线位置的链接。

### 1.8.0.1 关于 bibtex 的注释

模板默认使用的 bibtex 后端不能正确处理 Unicode 字符编码（即“国际”字符）。您可能会在编译日志中看到有关此问题的警告，如果您的参考文献包含 Unicode 字符，它们可能不会正确显示或根本不显示。解决这个问题一个办法是使用 biber 后端而不是过时的 bibtex 后端。这可以通过在 tex/in-header.tex 中找到 backend=bibtex 并将其更改为 backend=biber 来实现。对此进行一些 Google 搜索以找到更多信息。

## 1.9 论文特性和约定

为了充分利用这个模板，您可能需要遵循一些约定。

### 1.9.1 打印格式

这个论文模板是为双面打印（即页面的正面和背面都有内容）设计的，因为大多数论文都是这样打印和装订的。切换到单面打印很简单，只需在 \_quarto.yml 文件中的 classoptions: 中添加 "oneside"。页眉包含页面编号在外侧（这样容易翻到您想要的那一页）和章节名称在内侧。

文本默认设置为 11 点字体大小，单行间距，如果您愿意或需要，可以通过类选项调整文本大小和间距。可以通过将 "singlespacing" 替换为 "onehalfspacing" 或 "doublespacing" 来类似地更改间距。

### 1.9.2 使用 US Letter 纸张

模板中使用的纸张大小是 A4，这是欧洲的标准尺寸。如果您在其他地方使用这个论文模板，特别是在美国，您可能需要将 A4 纸张尺寸更改为 US Letter 尺寸。这可以通过编辑 `_quarto.yml` 中的 `geometry`：在 pdf 格式部分来实现。

### 1.10 表格

当您 `Quarto` 论文渲染为 PDF 时，它会很好地处理 `LaTeX` 表格代码。但是，如果您正在这样做，我猜您可能是在写 `LaTeX` 而不是 `Quarto` 论文。所以我不讨论 `LaTeX` 表格。相反，这里是如何使用 `R` 创建表格。`Python` 和 `Julia` 用户，您将有自己的表格包，但想法将相似。

表 1.1: 这是我的标题。

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4	21.0	6	160	110	3.90	2.875	17.02	0	1	4	4
Wag											
Datsun 710	22.8	4	108	93	3.85	2.320	18.61	1	1	4	1
Hornet 4	21.4	6	258	110	3.08	3.215	19.44	1	0	3	1
Drive											
Hornet	18.7	8	360	175	3.15	3.440	17.02	0	0	3	2
Sportabout											
Valiant	18.1	6	225	105	2.76	3.460	20.22	1	0	3	1

```
#| label: tbl-cars
#| tbl-cap: 这是我的标题。
knitr::kable(head(mtcars))
```

`#|` 是设置我们的交叉引用的方式，然后你可以像 `@tbl-cars` 那样引用表格。这是表 1.1。

有关完整示例和说明，请参阅 `Quarto` 手册。

## 1.11 图表

再次，我们用 Quarto (markdown) 而不是 LaTeX 写我们的图表。如果您真的想，您可以写 LaTeX，但它只会在 PDF 输出中被解释。

#| 是设置我们的交叉引用的方式，然后你可以像 @fig-cars 那样引用图表。有关完整示例和说明，请参阅 Quarto 手册。

### 1.11.1 排版数学

如果您的论文将包含大量的数学内容，LaTeX 将使其看起来非常漂亮，无论是 HTML 还是 PDF 输出。

[Not So Short Introduction to LaTeX](#) 应该告诉您大多数情况下排版数学所需知道的一切。如果您需要更多信息，AMS 提供了一个更全面的数学指南，名为 [A Short Math Guide to LaTeX](#)。

## 1.12 结束语

祝你好运，玩得开心！

本指南最初由 Sunil Patel 编写：<http://www.sunilpatel.co.uk> 和 Vel：<http://www.LaTeXTemplates.com> 并由 Eli Holmes 为 Quarto 严重缩短和改编。

## 第二章 Quarto 书稿速览

这是第二章的内容，概要性地讲解基于 quarto 拓展包进行图书排版的整体思路与实现方式。(Xie 2015; R-base?)

### 2.1 Quarto 的结构

Quarto 作为基于 Markdown 的写作工具，旨在简化和增强出版流程，它提供了一种灵活且易于使用的方式来创建和发布各种类型的文档。

Quarto 包括以下几个关键组成部分：

1. **Quarto Markdown 文件**：Quarto 使用 Markdown 作为主要的文档格式，文件扩展名为 `.qmd`。
2. **YAML 配置文件**：Quarto 使用 `_quarto.yml` 文件作为项目的配置文件。这个 YAML 配置文件就是一个文本文件，它允许用户指定输出格式、模板、外观等选项，自定义 Quarto 文档的元数据。
3. **输出格式**：Quarto 使用 **Pandoc** 转换文档格式，将 Markdown 文档输出为 HTML、PDF、ePub、docx 等格式，用户可以根据需要配置相应选项，选择输出格式。
4. **模板**：Quarto 使用模板来定义文档的整体布局、样式和体例等，用户可以选择预定义的模板，也可以自定义自己的模板，使用 HTML、CSS 或 SCSS、LaTeX 等方式来控制文档的形式。
5. **资源**：在 Quarto 中，你可以包含各种类型的资源，如图片、代码、数据、交互列表等，这些资源可以与 Markdown 文件一起存储在同一目录中，方便管理和引用。

通过将 Markdown 文件、YAML 配置文件、模板和各类资源整合到一起，Quarto 提供了一种完整的文档构建和发布解决方案，它使得创建专业、美观的多格式文档变得更加简单和高效。

Quarto 目前发行的版本为 1.4.554。

## 2.2 关于 Quarto Book

Quarto Markdown 作为新一代的 R Markdown, 是另一个增强 markdown 格式的扩展, 使得 qmd 格式可以支持书籍编写所需的章节、数学公式、定理、表格、图形、文献等浮动对象的自动编号, 并按设定的标签在正文中引用。在 quarto 的管理下一本书的内容可以按章节分解成多个 qmd 文件, 其中可以包含可执行的 R/Python/Julia/observable 等代码, 代码生成的统计汇总结果、表格、图形可以自动插入到生成的内容中, 而且其中的表格和图形也是浮动。书的输出格式包括支持 gitbook 格式的网页图书, 也可以经 LaTeX 编译器转换的 PDF 图书, 还可以生成 ePub 等格式的电子书。

目前 Quarto 支持在 RStudio、VS Code(甚至 Python Jupyter) 的集成环境中编辑、管理和生成图书。

关于浮动对象标签的设定与应用见第 2.6 节的汇总表格及后续各章的介绍与示例。

## 2.3 书的基本设置

一本用 bookdown 管理的书, 一般放置在某个子目录下, 并作为一个 RStudio 项目 (project) 用 RStudio 管理。该目录中的所有的文本文件都要使用 UTF-8 编码。

### 2.3.1 index.Rmd 文件

一本 Quarto 书, 一般都需要有一个 index.Rmd 文件, 这是最后生成的网站的主页的原始文件。这个文件的开始是 YAML 元数据部分, 进行全书的有关设置, 包括标题、作者、日期及影响全书的一些选项等, 放在三个减号组成的两行之间。然后写一些这本书的说明, 如书的前言部分。index.Rmd 中 YAML 元数据部分的一个例子如下:

```
title: "Quarto书稿模板"
author: "汤银才"
date: "2024-07-07"
documentclass: book
bibliography: [myrefs.bib]
biblio-style: apa
```

```
link-citations: yes
site: bookdown::bookdown_site
description: "bookdown 写书体验非常好."
```

注意:

1. `site` 选项很重要, 一定要有, `site: bookdown::bookdown_site` 使得 RStudio 能辨认这是一个 `bookdown` 图书项目, 从而为其生成一键编译的 `build` 快捷方式;
2. 在 `bookdown` 项目中与 `index.Rmd` 同级的所有 `.Rmd` 文件都自动作为书的一章, 其好处是作者可以任意地增删章节, 编译整本书时将按照文件名的字典序依次进行。实际上, 也可以在 `_output.yml` 文件中设置一项 `rmd_files`, 列出所有需要作为一章的文件, 并以列出次序编译;
3. 在 `index.Rmd` 的元数据中也可以指定一些 `LATEX` 的选项, 例如

```
fontsize: 12pt
indent: true
linestretch: 2.0
link-citations: yes
colorlinks: yes
lot: true
lof: true
geometry:
- a4paper
- tmargin=2.5cm
- bmargin=2.5cm
- lmargin=3.5cm
- rmargin=2.5cm
```

### 2.3.2 `_bookdown.yml` 文件

一个 `bookdown` 图书项目除了 `index.Rmd` 文件之外, 还有一些设置文件从 `index.Rmd` 文件的元数据部分抽离出来。一个是 `_bookdown.yml` 文件, 它存放与整本书的处理有关的 `YAML` 元数据。例如

```
book_filename: "bookdown-template"
new_session: yes
language:
  label:
    fig: "图 "
    tab: "表 "
    thm: '定理'
    def: '定义'
    exm: '例'
    proof: '证明: '
  ui:
    chapter_name: ["第 ", " 章"]
```

其中 `new_session: true` 设置很重要，这使得每一个 Rmd 文件中的 R 程序都在一个单独的 R 会话中独立地运行，避免了不同 Rmd 文件之间同名变量和同名标签的互相干扰。`book_filename` 是最终生成的 PDF 图书或者 ePub 电子书的主文件名。`language` 下可以定制一些与章节名、定理名等有关的名称。

### 2.3.3 `_output.yml` 文件

另一个设置文件是 `_output.yml`，用于图书输出格式的设置<sup>1</sup>，本小册子的 `_output.yml` 文件内容如下

```
bookdown::gitbook:
  css: css/style.css
  split_by: chapter
  includes:
    in_header: _header.html
  config:
    toc:
      collapse: subsection
      scroll_highlight: yes
    before: |
```

---

<sup>1</sup>这部分内容也可以包含在 `index.Rmd` 的元数据部分

```
<li><a href="./">bookdown排版模板</a></li>
after: |
  <li><a href="https://bookdown.org" target="blank">本书由 bookdown 强力驱动
download: [pdf, epub]
edit: https://github.com/yihui/bookdown-chinese/edit/master/%s
sharing:
  github: yes
  facebook: no
pandoc_args: [ "--cs1", "apa-6th-edition-no-ampersand.csl" ]
bookdown::pdf_book:
  includes:
    in_header: latex/preamble.tex
    before_body: latex/before_body.tex
    after_body: latex/after_body.tex
  keep_tex: yes
  dev: "cairo_pdf"
  latex_engine: xelatex
  citation_package: biblatex
  template: latex/template.tex
  toc_depth: 3
  toc_unnumbered: no
  toc_appendix: yes
  quote_footer: ["\\begin{flushright}", "\\end{flushright}"]
  pandoc_args: [ "--top-level-division=chapter" ]
bookdown::epub_book:
  stylesheet: css/style.css
  pandoc_args: [ "--cs1", "apa-6th-edition-no-ampersand.csl" ]
```

它分别对 `gitbook`、`pdf_book` 和 `epub_book` 三种输出格式设置了一些输出选项。其中一些选项是通过文件形式给出设置的，我们再补充说明一下。

1. `style.css` 是自定义的 CSS 显示格式，在 `gitbook` 和 `epub_book` 中使用；

2. `_header.html` 是插入了一部分个性化的 HTML 代码，其内容将出现在每个生成的 HTML 文件的 `head` 部分。我们在此文件中给出了使用本地的 Mathjax 实现数学公式离线显示的设置，内容为

```
<script type="text/x-mathjax-config">
MathJax.Hub.Config({
  jax: ["input/TeX","output/SVG"],
  extensions: ["tex2jax.js","MathMenu.js","MathZoom.js"],
  TeX: {
    Macros: {
      bm: [{"\\boldsymbol #1"},1],
    },
    extensions: ["AMSmath.js","AMSsymbols.js","noErrors.js","noUndefined.js"]
  }
});
</script>
<script type="text/javascript"
  src="http://127.0.0.1/MathJax/MathJax.js">
</script>
```

其中 `http://127.0.0.1/MathJax/` 是本地服务器上 Mathjax 的位置。有关 Mathjax 的本地安装与启动可参考第附录 B 章的介绍;

3. `apa-6th-edition-no-ampersand.csl` 是 gitbook 和 epub\_book 处理文献使用的风格文件;
4. `preamble.tex` 是处理 (编译)bookdown 文件经 pandoc 转化生成的 tex 文件时导言区需要额外的宏包和设置;
5. `before_body.tex` 是 tex 书稿类正文前面的设置, 最基本的是

`\frontmatter`

6. `after_body.tex` 是 tex 书稿类正文之后的设置, 最基本的是

`\backmatter`

7. ``template.tex`` 是针对 ``bookdown`` 编译经 ``pandoc`` 转化生成的 ``tex`` 文件时的模板, 由它生

其他选项说明:

1. `split_by: chapter`: 按章分割书稿;
2. `collapse: subsection`: 目录中隐藏子节 (仅显示二级标题);
3. `scroll_highlight: yes`: 目录滚动时高亮显示, 便于定位;
4. `keep_tex: yes`: 保留中间生成的 `tex` 源文件, 便于查错;
5. `dev: "cairo_pdf"`: 使用 `cairo_pdf()` 生成 `LaTeX` 编译需要的图片文件;
6. `latex_engine: xelatex`: `TeX` 文件的排版引擎为 `XeTeX`, 针对 `UTF-8` 编码;
7. `citation_package: biblatex`: 文献引用库指定为 `biblatex`, 另一个为 `natbib`;
8. `toc_depth: 3`: 目录提取至三级标题;
9. `toc_unnumbered: no`: 指定目录编号;
10. `toc_appendix: yes`: 附录添加到目录中.

## 2.4 章节结构

如前所述, 除了 `index.Rmd` 文件, 项目中每个 `.Rmd` 文件都作为一章, 其第一行是以一个 `#` 号和空格开头的一级标题。

每一章可以有若干节与子节, 分别用 `markdown` 的二级标题 (二个 `#` 开始) 和三级标题 (三个 `#` 开始) 编写。`bookdown` 的章、节、子节标题单独成一行, 其后可以添加标签, 章节的标签是标题后加空格, 然后是大括号内以 `#` 号开头的标签, 如

```
# 引言 {#intro}
```

```
## 关于bookdown {#bookdown}
```

`bookdown/quarto markdown` 中有二个特殊的标题:

## 1. 部分

内容相近的章节可以作为一个”部分”。为此，在一个部分的第一个章节文件的章标题前面增加一行，以 # (PART) 开头，以 {-} 结尾，例如

```
# (PART) bookdown 中的浮动对象 {-}
```

## 2. 附录

一本书的最后可以有附录，附录的章节将显示为 A.1, B.1 这样的格式。为此，在附录章节的第一个文件开头加如下的第一行标题行：

```
# (APPENDIX) 附录 {-}
```

```
# biblatex 介绍 {#biber}
```

## 2.5 书的编译

在 `index.Rmd` 或者 `_bookdown.yml` 中设置 `site: bookdown::bookdown_site` 后，RStudio 就能识别这个项目是一个 bookdown 项目，这时 RStudio 会有一个 Build 按钮，其中有 Build book 快捷图标，从下拉菜单中选择一个输出格式（包括 `gitbook`、`pdf_book`、`epub_book`），就可以编译整本书（见图 2.1）。

经 build 编译生成的图书默认保存在 `_book` 子目录中<sup>2</sup>。

1. 对 `gitbook` 格式（即 HTML 网页格式），编译完成后会弹出一个预览窗口，点击“Show in new window”按钮可以将内容在操作系统默认的网络浏览器中打开。我们也可以用其他浏览器（建议使用 Google chrome 浏览器）打开 `_book` 子目录中的 `index.html` 文件来查看 `gitbook` 格式的图书。
2. 对于 `pdf_book` 格式，如果成功编译<sup>3</sup>，也会弹出一个 PDF 预览窗口。可以在 `_book` 子目录中找到这个 PDF 文件。
3. 对于 `epub_book` 格式，如果成功编译，会在操作系统默认的 ePub 软件（如苹果电脑的 `book`）中打开，并在 `_book` 子目录中找到这个 ePub 文件。

<sup>2</sup>可以在 `_bookdown.yml` 中设置 `output_dir` 项改变图书保存的子目录。

<sup>3</sup>需要 L<sup>A</sup>T<sub>E</sub>X 支持，建议安排 TeXLive，也可以仅安装谢益辉为 `rmarkdown` 开发的 `tinytex`

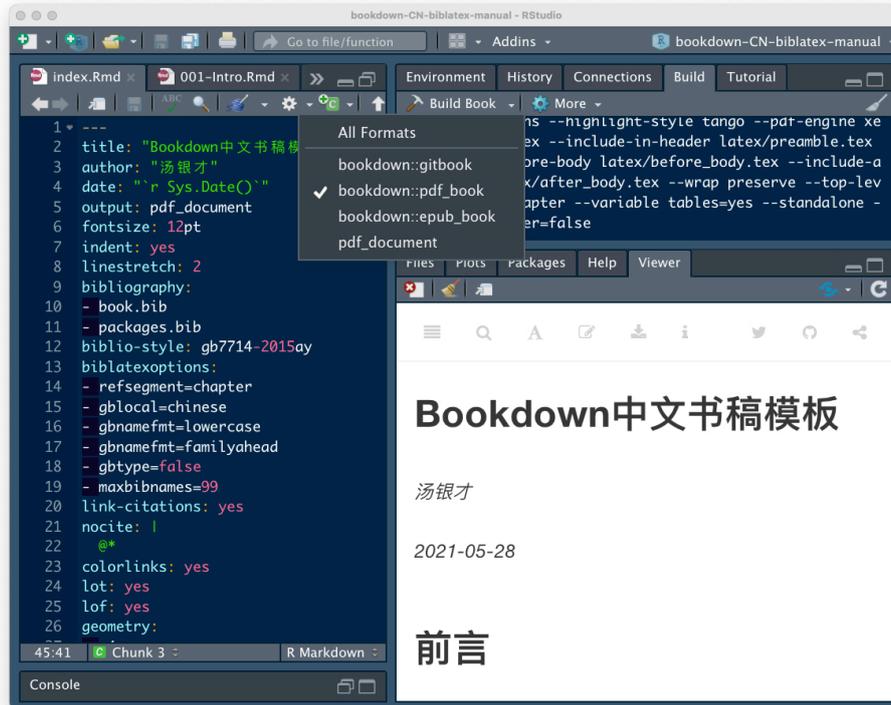


图 2.1: R Bookdown 编译界面.

## 2.6 浮动对象标签与引用汇总

浮动对象	标签设置	引用格式
标题	{#sec-label}	@sec-label
公式	{#eq-label}	(@eq-label)
图形	label="label"	\@ref(fig:label)
表格	label="label"	\@ref(tab:label)
定理	label="label"	\@ref(prefix:label)
文本	(ref:label)	`(ref:label)`
文献	biblabel	@biblabel

注:

1. 定理泛指定理类, 包括定理 (thm)、引理 (lem)、推论 (cor)、命题 (prp)、设想 (cnj)、定义 (def)、例子 (exm)、习题 (exr) 等, 其中括号中是引用时的前缀 (prefix);

2. 文本标签在单独一行中设定，可用在表格与图形的 `caption` 中引用，即在 `fig.caption`, `tab.caption` 选项的设置中引用；
3. 定理类环境标签前缀的汉化可在 `_bookdown.yml` 中通过 `language` 进行<sup>4</sup>，例如

```
language:  
  label:  
    fig: "图 "  
    tab: "表 "  
    thm: '定理'  
    def: '定义'  
    exm: '例'
```

---

<sup>4</sup>图形与表格也可同理汉化

## 第三章 Quarto 书稿中的章节标题

我们在第 三 章讲述章节标题的设置、标签与引用.

### 3.1 章节标题

章节标题用遵从 markdown 的规则, 用 # 设置,

- 一级标题用一个井号 #, 在 quarto 书稿中表示章, 相当于 TeX 中的 `\chapter{}`
- 二级标题用二个井号 ##, 在 quarto 书稿中表示节, 相当于 TeX 中的 `\section{}`
- 三级标题用三个井号 ###, 在 quarto 书稿中表示子节, 相当于 TeX 中的 `\subsection{}`

还可以有更深的标题设置.

### 3.2 章节标题标签的设定与引用

Quarto 书稿中章节标题标签可在标题后用 `{#sec-label}` 来设定, 引用方式为 `@sec-label`. 例如

```
第 @sec-sections 章 第 @sec-sec3-2 节讨论标题标签的设定与引用.
```

显示为:

第 三 章第 3.2 节讨论标题标签的设定与引用.



## 第四章 Quarto 书稿中的公式与定理

这是第 四 章的内容, 讲述浮动对象定理与公式的标签与引用.

### 4.1 公式标签的设定与引用

Quarto 借助 `mathjax`(或 `katex`) 处理数学公式的渲染; 尽管可通过联网由 `cdn` 上的 `mathjax.js` 进行渲染, 但速度随因公式的增加, 渲染变得很慢, 甚至出错。`mathjax` 的本地化是提速的主要解决方案, 详见附录 B 介绍;

#### 4.1.1 `{#eq-label}` 标签

Quarto 中无论是单行公式还是多行公式, 均通过公式无标号的公式环境 (一对 `$$`) 实现, 其后添加标签, 格式为 `{#eq-label}`, 其中 `eq` 是关键字, 例如

```
$$
\left(k\right) = \binom{n}{k} p^k \left(1-p\right)^{n-k}
$$ {#eq-binom}
```

显示为单行公式

$$f(k) = \binom{n}{k} p^k (1-p)^{n-k} \quad (4.1)$$

对于多行公式可以采用 `aligned` 环境, 但目前仅可对整个公式组设置标签, 例如

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

显示为

$$g(X_n) = g(\theta) + g'(\tilde{\theta})(X_n - \theta)$$

$$\sqrt{n}[g(X_n) - g(\theta)] = g'(\tilde{\theta})\sqrt{n}[X_n - \theta]$$

在这种情形下, 公式的引用采用 `@eq-label`, 例如上面的二个公式可引用为: 公式 (公式 4.1) 和公式 (公式 4.2).

#### 4.1.2 `\label` 标签

- 为了和 LaTeX 一样使用 `\label` 和 `\ref` 或 `\eqref` 去设定与引用公式标签, 我们需要在上面创建的 `mathjax.html` 文件中, 添加如下脚本:

```
<script>
MathJax = {
  tex: {
    tags: 'ams', // should be 'ams', 'none', or 'all'
  }
};
</script>
```

- 我们可以将 `equation` 和 `aligned` 环境结合起来输入多行公式, 对整个公式组设置一个 `\label`, 例如:

```
\begin{equation}
\begin{aligned}
g(X_{n}) &= g(\theta)+g'(\tilde{\theta})(X_{n}-\theta)\backslash\backslash
\sqrt{n}[g(X_{n})-g(\theta)] &= g'\left(\tilde{\theta}\right)\sqrt{n}[X_{n}-\theta ]
\end{aligned}\label{eq-1}
\end{equation}
```

显示为

$$g(X_n) = g(\theta) + g'(\tilde{\theta})(X_n - \theta)$$

$$\sqrt{n}[g(X_n) - g(\theta)] = g'(\tilde{\theta})\sqrt{n}[X_n - \theta] \tag{4.2}$$

- 我们还可以用 `align` 等环境输入多行公式，每行公式设置一个 `\label`，例如：

```
\begin{align}
a^n &= a \cdot a \cdot a \cdot \dots \label{eq-2} \\
a^n &= p \label{eq-3}
\end{align}
```

显示为

$$a^n = a \cdot a \cdot a \cdot \dots \quad (4.3)$$

$$a^n = p \quad (4.4)$$

在这种情形下，我们可以用和 TeX 一样的方式来引用公式，引用方式为 `\eqref{label}`，例如：公式 (4.2) 和 (4.4)。

### **i** 注释

`{#eq-label}` 和 `\label` 标签有各自适用的应用场景。

1. 在 Markdown 转换 (render) 为 PDF 时, Quarto 先将 Markdown 转换为 TeX, 再利用 TeX 系统编译输出 PDF. 在此过程中, `{#eq-label}` 标签被自动转换为 `\label` 标签, 因此两种标签会在 PDF 中合并编号;
2. 在 Markdown 转换 (render) 为 HTML 时, `{#eq-label}` 标签会转换为静态的 `\tag` 标签, 而 `\label` 标签保持不变, 因此会出现两种标签相互独立、分别编号的情况;
3. 在 Markdown 转换 (render) 为 ePub 时, `{#eq-label}` 标签会被保留, 而 `\label` 标签无法显示.

因此, 我们不建议混用 `{#eq-label}` 和 `\label` 这两种标签, 尤其在多格式出版中, 为尽量保证不同格式文件内容的一致性, 应优先考虑使用

`{#eq-label}` 标签, 暂时舍弃 `\label` 标签以及多行公式的逐行引用, 将这个问题留给 Quarto 在将来的升级版本中解决.

## 4.2 定理标签的设定与引用

这里我们给出几个 Quarto 书稿中定理类环境的例子.

引理 4.1. 带标签的引理.

定理 4.1 (无限群). 带标签与名字的定理.

证明. 这里是定理的证明部分. □

定义 4.1. 带标签的定义.

例 4.1. 带标签的例子.

定理类引用的示例: 例 4.1, 定义定义 4.1 定理定理 4.1.

## 4.3 数学公式的扩展

有些公式无法用 TeX 中包的命令来实现, 例如粗体数学符号, 尽管在 TeX 中有个 `bm` 包在数学环境下通过 `\bm{\alpha}` 来实现 `\boldsymbol{\alpha}` 的功能, 但在 html 下需要给 mathjax 做个 TeX 宏 (macro)`bm`<sup>1</sup>:

```
TeX: {
  Macros: {
    bm: [{"{\boldsymbol #1}"}],
  },
}
```

此时由  `$\bm{\alpha}$` , 出来的效果为  $\alpha$ .

有关数据公式的标签与应用可参考 [mathjax 官方文档](#), Mathjax 2.7 的本地化安装参考附录 B 介绍.

<sup>1</sup>配置在 `MathJax.Hub.Config` 下进行, 具体参见 Mathjax 技术文档说明

## 第五章 Quarto 书稿中的图形的处理

第 五 章将通过具体的示例展示浮动对象图形的标签与引用.

### 5.1 由 R 生成单个图形示例

这是最为常见的情形, 只涉及一个 R 的绘图主命令.

```
#| label: fig-fig5-1
#| fig: TRUE
#| echo: FALSE
#| fig.cap: "`iris`数据集`Petal.Length ~ Species`的箱线图."
boxplot(Petal.Length ~ Species, data = iris,
         ylab = "Petal length", xlab = "Species", col = "gray")
```

显示为

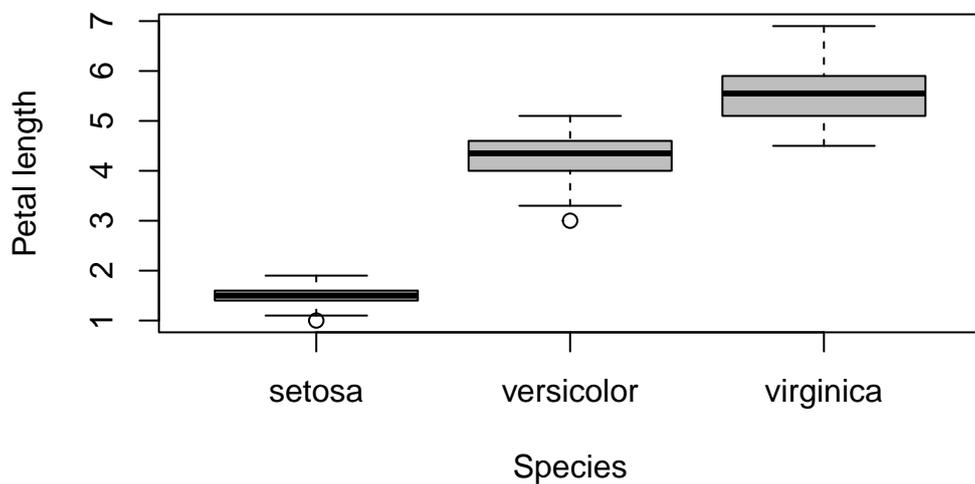


图 5.1: iris 数据集 `Petal.Length ~ Species` 的箱线图.

#### **i** 注释

图形的标签由 R 代码块中的 `label: fig-xxx` 给出, 其中 `fig` 是关键字.

## 5.2 由 R 生成两个图形并置或堆叠

在 R 的代码块选项中设置 `fig-subcap: true` 或 `fig-subcap` 加上子图的图例，并添加 `layout-ncol: 2` 就可获得二个图形的并置。例如下面的代码

```
#| label: fig-fig5-2
#| fig-cap: "`iris`数据集`Petal.Length ~ Species`的散点图。右侧的图像中散点类型通过
#| fig-subcap:
#|   - "散点图"
#|   - "添加回归线"
#| layout-ncol: 2
plot(Petal.Length ~ Petal.Width, data = iris)
plot(Petal.Length ~ Petal.Width, data = iris,
     pch = c(21, 8, 19)[as.numeric(iris$Species)])
legend("topleft", pch = c(21, 8, 19),
      legend = levels(iris$Species), box.lty = 0)
lm1 <- lm(Petal.Length ~ Petal.Width, data = iris)
abline(lm1) # 在散点图中添加拟合线
```

在 html 正常的浏览中得到两个左右并置的图形。

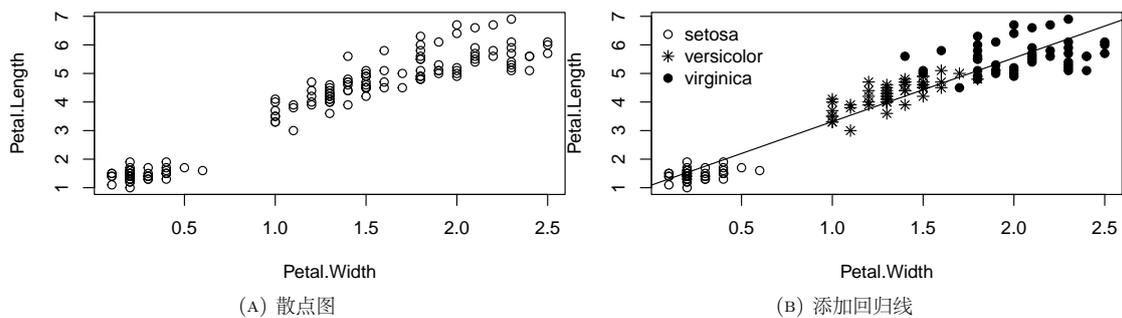


图 5.2: `iris` 数据集 `Petal.Length ~ Species` 的散点图。右侧的图像中散点类型通过 `Species` 因子的水平给出，见图例。直线为数据集拟合线性模型的结果。

类似地，通过选项 `layout-nrow: 2` 就可将二个图形堆叠起来。R 代码

```
#| label: fig-fig5-3
#| fig-cap: "`iris`数据集`Petal.Length ~ Species`的散点图。下图的图像中散点类型通过
#| fig-subcap:
```

```
#| - " 散点图"  
#| - " 添加回归线"  
#| layout-nrow: 2  
plot(Petal.Length ~ Petal.Width, data = iris)  
plot(Petal.Length ~ Petal.Width, data = iris,  
      pch = c(21, 8, 19)[as.numeric(iris$Species)])  
legend("topleft", pch = c(21, 8, 19),  
       legend = levels(iris$Species), box.lty = 0)  
lm1 <- lm(Petal.Length ~ Petal.Width, data = iris)  
abline(lm1)      # 在散点图中添加拟合线
```

在 html 正常的浏览中得到两个上下堆叠的图形.

### 5.3 单个静态图形的插入

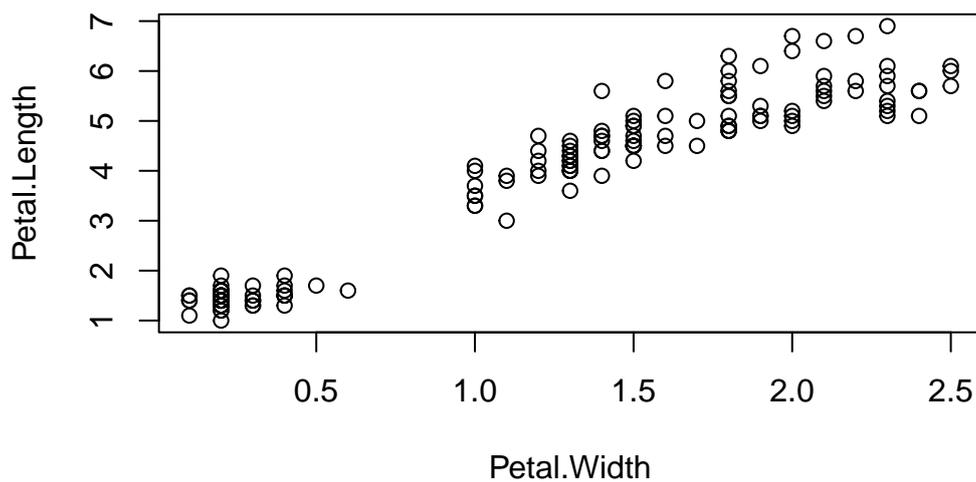
在 Quarto 中插入本地图形可使用命令. 代码

```
#| label: fig-fig4-4  
#| fig.align: 'center'  
#| out.width: '70%'  
#| fig.cap: 'R logo'  
knitr::include_graphics("../figures/Rlogo.png")
```

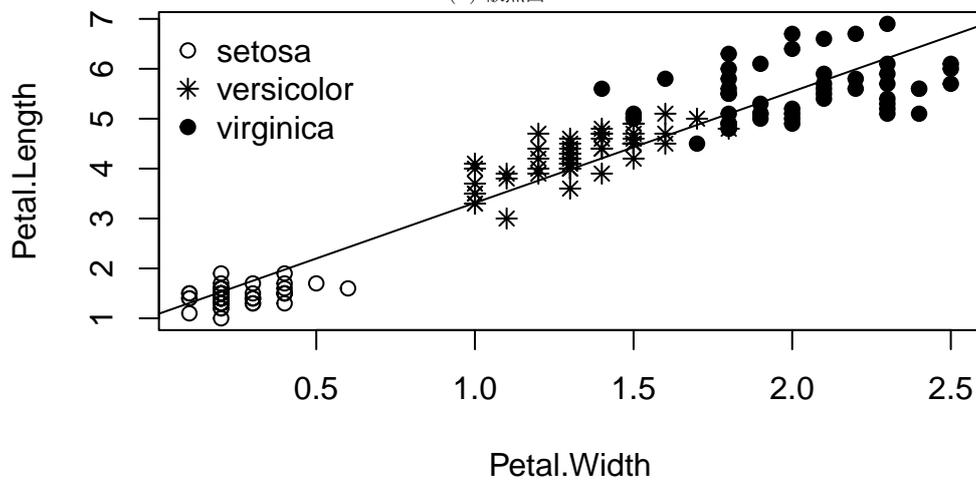
给出了 R 的 logo 图.

### 5.4 静态图形的并置

```
::: {#fig-program layout-ncol="3"}  
![R](../figures/Rlogo.png){#fig-R}  
  
![Python](../figures/python.png){#fig-python}  
  
![Julia](../figures/julia.png){#fig-julia}
```



(A) 散点图



(B) 添加回归线

图 5.3: iris 数据集  $Petal.Length \sim Species$  的散点图. 下图的图像中散点类型通过  $Species$  因子的水平给出, 见图例. 直线为数据集拟合线性模型的结果.



图 5.4: R logo

数据科学的编程语言

⋮

得到 R, Python 和 Julia 三个 logo 的并置.

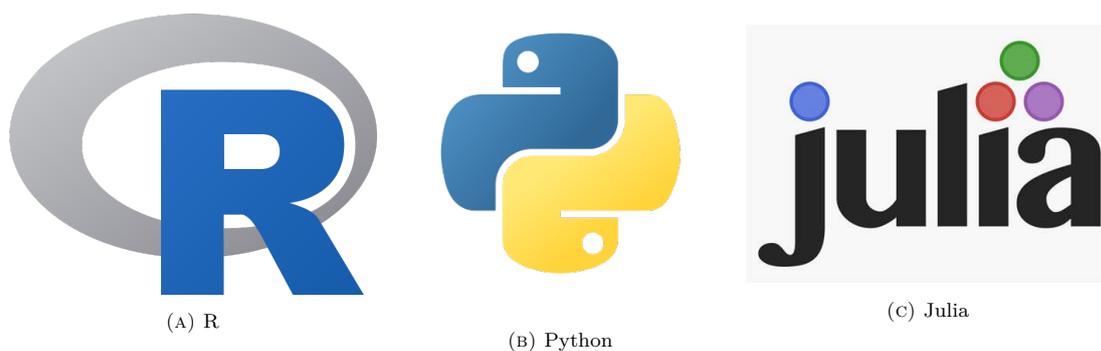


图 5.5: 数据科学中的编程语言

### **i** 注释

静态图形的插入我们可以使用 `knitr` 程序包中的命令 `include_graphics()`. 例如上面的代码等价于

```
knitr::include_graphics("../figures/Rlogo.png")  
knitr::include_graphics("../figures/python.png")  
knitr::include_graphics("../figures/julia2.png")
```

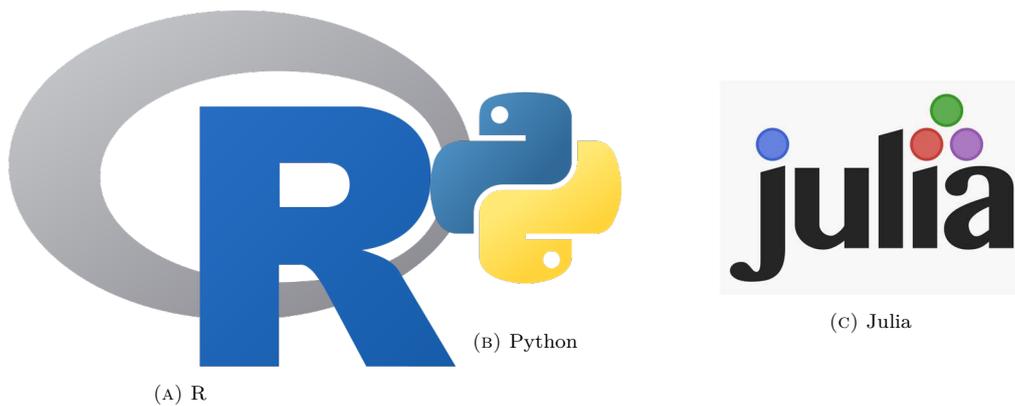


图 5.6: 数据科学中的编程语言

## 5.5 图形引用

图形引用通过 R 代码块的标签引用, 并带前缀 `fig-`, 例如

图 `@fig-fig5-2` 和图 `@fig-fig5-3` 为两个图的并置与堆叠.

输出为:

图图 5.2 和图图 5.3 为两个图的并置与堆叠.

## 第六章 Quarto 书稿中的表格的处理

第 六 章将通过具体的示例展示浮动对象“表格”的标签与引用技巧.

### 6.1 表格示例 1: 由数据生成表格

代码

```
#| label: tbl-mtcars
#| tbl-cap: "mtcars 数据的前 5 行."
library(knitr)
kable(head(mtcars[, 1:5], 10))
```

输出为

表 6.1: mtcars 数据的前 5 行.

	mpg	cyl	disp	hp	drat
Mazda RX4	21.0	6	160.0	110	3.90
Mazda RX4 Wag	21.0	6	160.0	110	3.90
Datsun 710	22.8	4	108.0	93	3.85
Hornet 4 Drive	21.4	6	258.0	110	3.08
Hornet Sportabout	18.7	8	360.0	175	3.15
Valiant	18.1	6	225.0	105	2.76
Duster 360	14.3	8	360.0	245	3.21
Merc 240D	24.4	4	146.7	62	3.69
Merc 230	22.8	4	140.8	95	3.92
Merc 280	19.2	6	167.6	123	3.92

#### **i** 注释

表格的标签由 R 代码块中的 `label: tbl-xxx` 给出, 其中 `tbl` 是关键字.

## 6.2 表格示例 2: 由数据框构造表格

代码

```
#| label: tbl-tab3-2
#| tbl-cap: '遗传连接模型例子中观测到的频数  $y_i$  和频率  $p(y_i|\pi)$ ,  $i=1, \dots, 4$ 
#| echo: FALSE
#| results: "asis"
tab.pis <- data.frame(
  C = c("Frequency  $y_i$ ", "Probability  $p(y_i|\pi)$ "),
  V1 = c("125", " $\frac{1}{2} + \frac{\pi}{4}$ "),
  V2 = c("18", " $\frac{1}{4}(1 - \pi)$ "),
  V3 = c("20", " $\frac{1}{4}(1 - \pi)$ "),
  V4 = c("34", " $\frac{1}{4}$ ")
)
names(tab.pis) <- c("Category", "1", "2", "3", "4")

knitr::kable(tab.pis, escape = FALSE)
```

输出为

表 6.2: 遗传连接模型例子中观测到的频数  $y_i$  和频率  $p(y_i|\pi)$ ,  $i = 1, \dots, 4$ , 197 个动物.

Category	1	2	3	4
Frequency $y_i$	125	18	20	34
Probability $p(y_i   \pi)$	$\frac{1}{2} + \frac{\pi}{4}$	$\frac{1}{4}(1 - \pi)$	$\frac{1}{4}(1 - \pi)$	$\frac{1}{4}$

### **i** 注释

表格中的题图通过 `tbl-cap`: 由文本引用生成.

## 6.3 表格示例 3: 由 `markdwon` 表格直接生成

INLA 可处理的分布/似然有 60 种, 表 6.3 仅列出了其中的一部分.

```

| **Value** | **Likelihood** |
|-----|-----|
| `poisson` | Poisson |
| `binomial` | Binomial |
| `t` | Student's t |
| `gamma` | Gamma |

```

: `INLA` 提供的一些似然. `{#tbl-likes}`

输出为

表 6.3: INLA 提供的一些似然.

Value	Likelihood
poisson	Poisson
binomial	Binomial
t	Student's t
gamma	Gamma

### **i** 注释

1. 表格中% 用\\%
2. 表格中 latex 命令用\\代替\
3. 前二种方法中表格的标签由 R 代码块中的'label:' 给出.

## 6.4 markdown 子表的并置

```

::: {#tbl-panel layout-ncol="2"}
| Col1 | Col2 | Col3 |
|-----|-----|-----|
| A | B | C |
| E | F | G |
| A | G | G |

```

表 6.4: 二个子表的并置

(A) 第一个表格			(B) 第二个表格		
Col1	Col2	Col3	Col1	Col2	Col3
A	B	C	A	B	C
E	F	G	E	F	G
A	G	G	A	G	G

```
: 第一个表格 {#tbl-first}
```

```
| Col1 | Col2 | Col3 |
|-----|-----|-----|
| A    | B    | C    |
| E    | F    | G    |
| A    | G    | G    |
```

```
: 第二个表格 {#tbl-second}
```

```
二个子表的并置
```

```
...
```

输出为

## 6.5 数据子表的并置与堆叠

在 R 的代码块选项中设置 `tbl-subcap: true` 或 `tbl-subcap` 加上子表的图例，并添加 `layout-ncol: 2` 就可获得二个子表的并置.

```
#| label: tbl-tables
#| tbl-cap: " 两个表格并置示例 2"
#| tbl-subcap:
#|   - "Cars 数据"
#|   - "Pressure 数据"
#| layout-ncol: 2
```

表 6.5: 两个表格并置示例 2

(A) Cars 数据	(B) Pressure 数据
speed    dist	temperature    pressure
4      2	0      0.0002
4      10	20     0.0012
7      4	40     0.0060
7      22	60     0.0300
8      16	80     0.0900
9      10	100    0.2700

```
library(knitr)
kable(head(cars))
kable(head(pressure))
```

输出为

类似地, 通过选项 `layout-nrow: 2` 就可将二个图堆叠起来.

```
#| label: tbl-tables-by-row
#| tbl-cap: " 两个表格堆叠示例"
#| tbl-subcap:
#|   - "mtcars 数据"
#|   - "iris 数据"
#| layout-nrow: 2

library(knitr)
kable(head(mtcars))
kable(head(iris))
```

输出为

## 6.6 表格引用

表格引用由代码块的标签 (设为 `label`) 引用实现, 并带前缀 `tbl-`, 由 `@tbl-label` 实现. 例如

表 6.6: 两个表格堆叠示例

(A) mtcars 数据

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225	105	2.76	3.460	20.22	1	0	3	1

(B) iris 数据

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
5.4	3.9	1.7	0.4	setosa

本章共出现二个表格并置的示例，即 表 @tbl-panel，表 @tbl-tables 和 @tbl-second.

输出为：

本章共出现二个表格并置的示例，即表表 6.4, 表表 6.5 和表 6.4b.

## 附录 A 附录标题

加油! 终于到最后一个部分了!

### A.1 附录中的图形、表格、公式

附录中的公式 (A.1) 和 (A.2) 分别为:

$$c = S_0 N(d_1) - X e^{-rT} N(d_2) \quad (\text{A.1})$$

和

$$p = X e^{-rT} N(-d_2) - S_0 N(-d_1), \quad (\text{A.2})$$

### A.2 R 代码

- 线性回归

```
par(mar = c(4, 4, 1, .1))
fit = lm(dist ~ 1 + speed, data = cars)
plot(cars, pch = 19, col = 'blue', las = 1)
abline(fit, lwd = 2)
```

- ggplot2

```
par(mar = c(4, 4, 1, .1))
fit = lm(dist ~ 1 + speed, data = cars)
plot(cars, pch = 19, col = 'blue', las = 1)
abline(fit, lwd = 2)
```

### A.3 Python 代码

```
import matplotlib.pyplot as plt
import numpy as np
x = np.arange(0.0, 6.0, 0.01)
```

```
plt.plot(x, [x**2 for x in x])  
plt.show()
```

## 附录 B Mathjax 的离线安装与使用

附录 B 介绍了网页显示数学公式的插件 mathjax，本地化安装和使用方法等. (Xie 2015; **R-base?**)

### B.1 mathjax 简介

- **MathJax**是一款相当强悍的在网页显示数学公式的插件.
- 基于 **Mathjax**, 就可通过 LaTeX 的命令输出精美的数学公式. 加载 **Mathjax** 后<sup>1</sup>, 就可通过一对美元符号  $(或左\右 \)$  输入行内公式, 通过一对双美元符号  $$(或左\[右\])$  输入行间公式, 例如

$$$$

```
J\alpha(x) = \sum_{m=0}^{\infty} \frac{(-1)^m}{m! \Gamma(m + \alpha + 1)} \left(\frac{x}{2}\right)^{2m + \alpha}
```

$$$$

显示出下面的数学公式

$$J\alpha(x) = \sum_{m=0}^{\infty} \frac{(-1)^m}{m! \Gamma(m + \alpha + 1)} \left(\frac{x}{2}\right)^{2m + \alpha}.$$

- 可以使用 LaTeX 自带的复杂的数学环境和数学字体命令, 如排版多行公式的 `align` 和 `split` 环境, 字母加黑命令 `\mathbb{}`

```
\begin{align}
3x-1 &= \mathbb{A} \\
3x &= \mathbf{B} \\
x &= \mathscr{C}
\end{align}
```

---

<sup>1</sup>需要远程或本地支持

输出为

$$3x - 1 = A \quad (\text{B.1})$$

$$3x = B \quad (\text{B.2})$$

$$x = C \quad (\text{B.3})$$

## B.2 调用远程服务器上的 mathjax

一般情况下, 只需要使用远程加载 Mathjax 的 js 库就行了, 例如在需要渲染数学公式的网页上增加 html 命令

```
</script>
<script type="text/javascript" async
  src="https://cdn.mathjax.org/mathjax/latest/
  MathJax.js?config=TeX-MML-AM_CHTML">
</script>
```

## B.3 mathjax 本地服务器的安装与使用

我们以 Macbook 的 Apache 服务器为例说明步骤<sup>2</sup>

### 1. 服务器的启动

在终端 (terminal) 下输入命令

```
sudo apachectl start
```

### 2. 检查服务是否启动成功

在浏览器中输入网址

```
http://127.0.0.1/
```

如果显示 It Works! 就表示服务器已经成功启动. 请记住: 服务器上文件在本地的位置为

```
/Library/WebServer/Documents
```

<sup>2</sup>Windows 10 下的本地服务器的启动可参考 (<https://www.jianshu.com/p/d86c77942181>)

### 3. 关闭服务器 (不用时)

在终端 (terminal) 下输入命令

```
sudo apachectl stop
```

### 4. 将 Mathjax2.6 或 Mathjax2.7 下载并解压到/Library/WebServer/Documents 下<sup>3</sup>, 目录名为 Mathjax

### 5. 启动本地 Mathjax

在运行 Bookdown(或其他 R Markdown, Quarto Markdown 文件) 时, 须加载下面的由 html 命令组成的文件 mathjax\_27.html

```
</script>  
<script type="text/javascript"  
  src="http://127.0.0.1/MathJax/MathJax.js">  
</script>
```

并在 Bookdown 或 Quarto Markdown 中由 `_output.yml` 的命令

```
include-in-header: mathjax_27.html
```

加载进来.

---

<sup>3</sup>暂且不要用最新的 3.2 版本



## 附录 C 基于 biblatex 生成参考文献

附录 C 介绍了文献库扩展包 `biblatex` 的使用，及其与 `natbib` 的比较.

### C.1 概述

- `bibtex` 与 `biber` 是处理参考文献信息的二个外部程序 (backend), 它们起到将 LaTeX 文档与 `bib` 文档交互作用的目的;
- `natbib` 与 `biblatex` 是二个处理参考文献 (bibliography) 展示和引用 (citation) 的 TeX 宏包; `natbib` 仅通过 `bibtex` 起作用, 而 `biblatex` 可通过 `biber` 起作用;

#### C.1.1 `natbib` 的优缺点

- `natbib` 的优点:
  - 有大量与期刊/出版商对应的风格文件 (.sty);
- `natbib` 的缺点:
  - 自定义的局限: 修改风格文件较为困难;
  - 设计的局限性: 主要为自然科学类期刊的 Author-Year 及数字引用方式设计.

#### C.1.2 `biblatex` 的优缺点

- `biblatex` 被认为是 LaTeX 中处理参考文献 (bibliography) 很有前途的宏包, 功能强大, 提供了很多可定制的选项.
- `biblatex` 的优点
  - 提供 `natbib` 的 Author-year 和数字引用方式, 因此可视为 `natbib` 的扩展;
  - 可通过 LaTeX 宏来控制文献的风格, 方便修改;
  - 提供许多人性化的引用风格 (例如 `author-title`);
  - 提供人性化的文献数据库域名 (字段);

- 直接支持多文献库与文献分类;
  - 提供大量标准与拓展的 `biblatex` 风格 (见使用手册);
  - 文献可按主题分割为部分.
- `biblatex` 的优点
    - 学习曲线陡: 选项多, 不易掌握;
    - 编译时可能会依赖于最新 LaTeX 版本, 从而导致版本的更新, 由此带来不必要的麻烦.

## C.2 `biblatex` 的在 TeX 中的使用样例

### 1. 使用示例

```
\usepackage[style=authoryear,
             backend=biber]{biblatex}
```

代替 `\usepackage[authoryear]{natbib}`;

### 2. 加载一个或多个 bib 文献库文件

```
\addbibresource{file1.bib}
\addbibresource{file2.bib}
```

### 3. 文献目录输入: 出现文献的地方输入

```
\printbibliography
```

### 4. 引用:

使用 `\textcite` 代替 `\citet`; 使用 `\parencite` 代替 `\citep`

### 5. 编译方式, 以 XeTeX 引擎为例

```
XeLaTeX
biber (代替 bibtex)
XeLaTeX
XeLaTeX
```

### C.3 biblatex 在 Quarto 书稿中的使用

1. 在 `index.qmd` 文件的 `yml` 部分增加选项

```
biblatexoptions: [refsegment=chapter]
biblio-style: gb7714-2015ay
```

注意 `[]` 内可增加其他 `biblatex` 选项 (参考 `biblatex` 使用手册)

2. 在需要出现文献的地方 (如每一章后) 加

```
\printbibliography[segment=\therefsegment,
                    heading=subbibliography,
                    title={参考文献}]
```

3. 在 `_output.yml` 文件的 `bookdown::pdf_book:` 增加选项 (前面空二格)

```
citation_package: biblatex
```

#### **i** 注释

若要 `bibtex` 驱动文献的排版, 只需要在这一步改为 `citation_package: natbib`

4. 如上所述, LaTeX 要生成最终的 PDF 文档. 如果论文中含有交叉引用 (图形、表格、公式、章节、文献等)、`bibtex/biber` 命令和术语表等, 通常需  
要让 TeX 编译多次. 而使用 `Latexmk`, 只需要让 TeX 编译一次, 它会自动  
帮你做好其它所有事情. 由于系统在你已经安装的 LaTeX 发行版本时已经  
包含了 `latexmk`, 但我们需要人为地运行 `Latexmk` 命令 (编译 TeX 文件),  
使用 XeLaTeX 的编译格式为

```
latexmk -pvc -xelatex file.tex
```

其中选项 `pvc` 表示检查输入文件的更改并预览结果. 在 `Rstudio` 中你只需要添加下面的代码块.

```
Sys.setenv(RSTUDIO_PDFLATEX = "latexmk")
```

## 参考文献

- Xie, Yihui. 2015. *Dynamic Documents with R and Knitr*. 2nd ed. Boca Raton, Florida: Chapman; Hall/CRC. <http://yihui.name/knitr/>.
- 茆诗松, and 汤银才. 2012. 贝叶斯统计分析. 北京: 中国统计出版社.
- 韦来生, and 张伟平. 2022. 贝叶斯分析. 北京: 高等教育出版社.



## 在学期间的研究成果及发表的论文

### 在国际和国内学术刊物上发表的论文

Liu, Y., liu, Y., Fan, Y. and Han, H., 2018. Likelihood ratio confidence interval for the abundance under binomial detectability models[J]. *Metrika*, 81: 549–568.

Liu, Y., liu, Y., Li, P. and Qin, J., 2018. Full likelihood inference for abundance from continuous time capture-recapture data[J]. *Journal of the Royal Statistics Society: Series B*, 80: 995–1014.

### 待发表论文

Smith, J., 2020a. Title of the paper[Z/OL]. [2020-01-18]. <http://xxx.xxx.cn>.

Smith, J., 2020b. Title of the paper[Z].

注: 硕士毕业论文若没有发表的成果, 这部分可以删除!



## 致 谢

从 200x 年 9 月至今的三年学习期间得到了数理信息学院许多老师、同学和朋友的帮助, 在此一并表示感谢.

在论文的选题到完成的各个阶段, 自始至终得到了导师 xxx 的细心指导和帮助, 并提供了许多宝贵的资料和建议, 其..... 和严谨的治学精神是我整整三年学习期间最为珍贵的养份, 在此我想由衷地说一声: 谢谢 xxx 老师, 谢谢你给我的无私的帮助!

最后, 也是最为重要的, 我的 xxx 多年来一直支持我的学习和研究, 在论文的完成过程中付出了大量的时间和心血. 感激之情, 难以言表, 我将永身不忘.