

R 语言编程：基于 tidyverse

第 26 讲 R 语言与 Latex, Git/Github 交互

张敬信

2022 年 3 月 25 日

哈尔滨商业大学

一. R 与 Latex 交互

Latex 是高质量的专业排版系统，具有强大的数学公式排版功能，非常适合生成高印刷质量的科技和数学类文档。Latex 需要编写代码再编译成 pdf，缺点是不像 Word“所见即所得”，现已广泛应用于书籍、期刊论文、毕业论文、学术报告、简历等排版。

对于大多数普通用户来说，只专注于使用现成 Latex 模板即可，模板已包含了全部的文档格式，只需要替换成自己相应的内容。

R markdown 就是将 Latex 排版融入进来：Rmd->md->tex->pdf，最终输出 pdf 文档¹，顺便解决了“插入代码块，并能运行代码，将代码运行结果显示出来”。

¹R markdown 输出 pdf 文档的另一条路线是 pagedown 包：Rmd->md->html->pdf。

1. Latex 开发环境

Latex 的主流开发环境是 TexLive (2021 版安装包已达 4.1GB), 编辑器可以选用 Texwork, TexStudio、VScode 等。

对于 R 用户, 强烈建议使用谢益辉专为 R markdown 开发的, 超轻量级的 Latex 环境: TinyTex + RStudio.

TinyTex 只保留了编译 Latex 的最核心组件, 以及少量常用宏包, 大小只有二百来兆。对于使用过程中缺少的宏包, 根据需要自动下载安装。

为了方便地在 R 环境中使用 Latex, 谢益辉还开发了 tinytex 包, 里面提供了各种操作 Latex 的方便函数。

2. 安装 TinyTex

国内现下载安装基本不可能成功，建议先从

<https://yihui.org/tinytex/TinyTeX.zip>

下载到本地，比如 D 盘根目录，再用命令从本地安装。

```
library(tinytex)      # 在 Console 窗口执行
tinytex::install_prebuilt(pkg = "D:/TinyTeX.zip")
# tinytex::uninstall_tinytex()  # 卸载 TinyTex
```

参数 pkg 指定 zip 文件路径，还有参数 dir 可以设置安装路径。

安装成功后可查看：

```
tinytex_root()        # 查看安装路径
tl_pkgs()              # 查看已安装宏包
```

3. 基本使用

- 修改国内镜像源

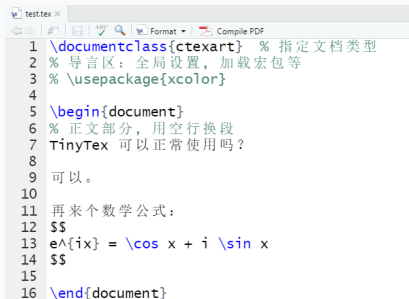
用的时候，不可避免需要下载宏包，所以先修改为国内镜像源：

```
# 修改清华大学的镜像源
```

```
tlmgr_repo(url="http://mirrors.tuna.tsinghua.edu.cn/CTAN/")
```

- 简单测试

在 RStudio 新建 Text File，输入 Latex 代码，保存的时候，后缀名用 .tex，即保存为 Latex 文件



```
1 \documentclass{ctexart} % 指定文档类型
2 % 导言区：全局设置，加载宏包等
3 % \usepackage{xcolor}
4
5 \begin{document}
6 % 正文部分，用空行换段
7 TinyTex 可以正常使用吗？
8
9 可以。
10
11 再来个数学公式：
12 $$
13 e^{ix} = \cos x + i \sin x
14 $$
15
16 \end{document}
```

英文 tex 文档一般用 pdf_latex 编译，中文 tex 文档特别是涉及使用系统自带中文字体，需要用 xelatex 编译。

运行:

```
xelatex("test.tex")
```

即用 xelatex 编译, 会自动下载安装所有缺少的宏包²。编译成功, 将在当前路径下, 生成 test.pdf。

至此, 已经成功搭建了 Latex 开发环境, 完全可以取代 Texlive 而使用, 也能编译.Rmd 到 pdf 文档。更多 Latex 语法参阅 [潘建瑜: LaTeX 科技排版入门](#)。

²除了缺少宏包, 有时候编译不成功, 可能是因为系统缺少字体, 需要手动下载再右键安装。

4. Latex 嵌入 Rmd

Latex 模板可以直接在 TinyTex + RStudio 开发环境使用，但是要嵌入到 Rmd 模板需要做一定的移植工作，同样对于大多数普通用户来说，会用别人移植好的模板就够了。

- 用 Latex 输入数学公式

要编译成 pdf 的 Rmd 文档中，可以使用 Latex 代码输入数学公式。行内数学公式， \dots ，行间数学公式用
$$\dots$$
。

\$\$

\begin{aligned}

\int_a^b f(x) \mathrm{d}x \approx \sum_{k=1}^n

\frac{h}{2} [f(x_{i-1}) + f(x_i)] \\\

\& = \frac{h}{2} [f(a) + f(b)] + h \sum_{k=1}^{n-1} f(x_i)

\end{aligned}

\$\$

$$\begin{aligned}\int_a^b f(x) \mathrm{d}x &\approx \sum_{k=1}^n \frac{h}{2} [f(x_{i-1}) + f(x_i)] \\ &= \frac{h}{2} [f(a) + f(b)] + h \sum_{k=1}^{n-1} f(x_i)\end{aligned}$$

一些控制 Latex 编译的选项可以在 yaml 中设置:

output:

pdf_document:

latex_engine: xelatex

citation_package: natbib

keep_tex: true

includes:

in_header: preamble.tex

before_body: doc-prefix.tex

after_body: doc-suffix.tex

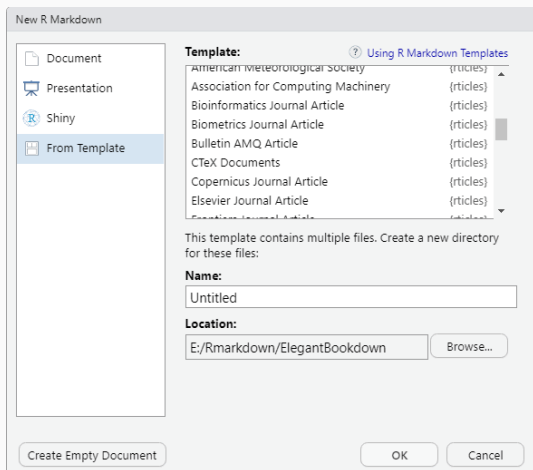
template: quarterly-report.tex

fontsize: 11pt

geometry: margin=1in

5. 期刊论文模板

安装 `rticles` 包后, Rmd 则多了很多可选的期刊模板:



6. 幻灯片模板

(1) xaringan 包

写轮眼包由谢益辉开发，安装后再从 From Template 选择模板时，可以选择：

- Ninja Presentation: 英文幻灯片
- Ninja Presentation (Simplified Chinese): 中文幻灯片

基本与普通 Rmd 一样的语法，用 "---" 换页。

点击 Knit 按钮编译到 HTML³。

³也可以为 RStudio 安装 Infinite Moon Reader (无限月读) 插件，在 Viewer 窗口实时预览幻灯片（保存则自动编译）。

(2) PPT 模板

新建 Rmd, 选择 Presentation -> PowerPoint, 或者从 From Template 选择 officedown 包提供的 Advanced PowerPoint Presentation 模板。

还可以使用 Office 自带的 PPT 模板, 作为自定义模板:

output:

```
powerpoint_presentation:  
  reference_doc: my-styles.pptx
```

点击 Knit -> Knit to PowerPoint 启动编译, 等待生成 PPT。

(3) R Beamer 模板

Beamer 是 Latex 下的一大类幻灯片模板, R markdown 已将其移植过来。新建 Rmd, 选择 Presentation -> PDF (Beamer), 开始使用。

Beamer 在 Beamer 主题矩阵页:

<https://hartwork.org/beamer-theme-matrix>

有大量的主题可选, 在 yaml 中设置主题名即可使用。

注意, 原始 R Beamer 模板只支持英文, 要想使用中文, 需要修改编译引擎为 xelatex, 并加载 ctex 宏包:

```
output:
  beamer_presentation:
    latex_engine: xelatex
    theme: "Madrid"
    colortheme: "dolphin"
    fonttheme: "structurebold"
header-includes:
  - \usepackage{ctex}
```
























点击 Knit -> Knit to PDF (Beamer) 启动编译, 等待生成 PDF。

7. 书籍模板

纯 Latex 书籍模板可以在 Latex 开发环境下运行，但不能包含并运行程序代码。谢益辉开发的 bookdown 包是 R Markdown 向书籍模板上的扩展，使得 Rmd 可以支持章节结构、公式图表、自动编号、交叉引用、参考文献等适用于编写书籍的功能。

可用的 bookdown 书籍模板：

- bookdown-demo：英文书籍（谢益辉）
- bookdown-chinese：中文书籍（谢益辉）
- bookdownplus 包：提供更多的 bookdown 书籍模板
- ElegantBookdown：移植 Elegant 书籍模板（黄湘云、叶飞）

名称	修改日期	类型	大小
 .gitignore	2018/10/22 9:36	GITIGNORE 文件	1 KB
 .travis.yml	2018/10/22 9:36	YML 文件	1 KB
 _bookdown.yml	2018/10/22 9:36	YML 文件	1 KB
 _build.sh	2018/10/22 9:36	SH 文件	1 KB
 _deploy.sh	2018/10/22 9:36	SH 文件	1 KB
 _output.yml	2018/10/22 9:36	YML 文件	1 KB
 01-intro	2018/10/22 9:36	RMD 文件	2 KB
 02-literature	2018/10/22 9:36	RMD 文件	1 KB
 03-method	2018/10/22 9:36	RMD 文件	1 KB
 04-application	2018/10/22 9:36	RMD 文件	1 KB
 05-summary	2018/10/22 9:36	RMD 文件	1 KB
 06-references	2018/10/22 9:36	RMD 文件	1 KB
 book.bib	2018/10/22 9:36	BIB 文件	1 KB
 bookdown-demo	2018/10/22 9:36	RPROJ 文件	1 KB
 DESCRIPTION	2018/10/22 9:36	文件	1 KB
 Dockerfile	2018/10/22 9:36	文件	1 KB
 index	2018/10/22 9:36	RMD 文件	2 KB
 LICENSE	2018/10/22 9:36	文件	7 KB
 now.json	2018/10/22 9:36	JSON 文件	1 KB
 preamble	2018/10/22 9:36	TEX 文件	1 KB
 README	2018/10/22 9:36	Markdown File	1 KB
 style	2018/10/22 9:36	层叠样式表文档	1 KB
 toc	2018/10/22 9:36	层叠样式表文档	3 KB

(1) 文件结构

整个书籍模板是一个 R 项目，书籍与期刊论文没有本质区别，只是因为结构更庞大，而被拆分成更多的文件，就涉及相互的串联。

书籍一般包含多章，每章是一个 Rmd 文件（必须 UTF-8 编码），每章章头是一级标题：# 章名；index.Rmd 里面包含部分 ymal 总是作为第 1 个，其他的默认按文件名顺序，当然也可以在 ymal 中定义顺序：

```
rmd_files:
```

- "index.Rmd"
- "01-intro.Rmd"
- "02-basic.Rmd"

在定义章节时，可以同时定义其交叉引用：

```
## 节标题 {#sec1}
```

```
...
```

```
\@{#sec1}
```

章节是默认编号的，若不编号需要：# Preface {-}

另外，.yaml 文件都是设置 ymal; .tex 文件都是需要在正文前加入设置和定义 Latex 相关；.bib 都是参考文献；相关的数据、脚本、图片都可以分别放在一个文件夹。

(2) 交叉引用

图、表、公式的交叉引用，这是 bookdown 包提供的，只支持 yaml 输出：

output:

```
bookdown::html_document2: default
```

```
bookdown::pdf_document2: default
```

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

图、表交叉引用，需要在相应的图、表代码块起名，再用 \@ref() 引用名字。

- 如图 \@ref(fig:cars-plot) 所示：

```
```{r cars-plot, fig.cap=" 汽车散点图", echo=FALSE}  
plot(cars)
```
```

- 见表\@ref(tab:mtcars).

```
```{r mtcars, echo=FALSE}
knitr::kable(mtcars[1:5, 1:5], caption = " 汽车数据")
```
```

数学公式，需要用 Latex 语法生成带编号的公式，同时对公式起名，再用\@ref() 引用

```
\begin{equation}
\bar{X} = \frac{\sum_{i=1}^n X_i}{n} (\#eq:mean)
\end{equation}
```

由式\@ref(eq:mean) 可得, ...

使用脚注, 先标记文字 ^[ftname], 再另起段: ^[ftname]: 脚注内容.

支持 bib 参考文献，所有参考文献放一起作为.bib 文件

```
@Book{zhaopeng2021,  
  title = {现代统计图形},  
  author = {赵鹏, 谢益辉, 黄湘云},  
  publisher = {人民邮电出版社},  
  address = {北京},  
  year = {2021},  
  edition = {1},  
  note = {},  
}
```

然后在正文里使用 [zhaopeng2021] 引用该文献。Zotero 软件可以很方便地管理参考文献，将文献批量导出到一个.bib 文件中。

(3) 环境与中文

Bookdown 提供了定理类环境: theorem, lemma, corollary, proposition, definition, example, exercise 等, 其交叉引用与引用公式类似, 需用到相应的缩写 thm, lem, cor, prp, def, exm, exr.

```
::: {.theorem #weakconv name=" 弱若收敛定理"}  
$\xi_n$ 依分布收敛到 $\xi$, 当且仅当对任意 $\mathbb{R}$ 上  
的一元实值连续函数 $f(\cdot)$ 都有  
$$  
E f(\xi_n) \rightarrow E f(\xi), \quad n \rightarrow \infty  
$$  
:::
```

然后这样引用: 由定理 \@ref(thm: weakconv) 得...

中文书籍，除了如前文所述设置 xelatex 引擎和中文字体之外，图、表、章节标题、定理类关键字也要改成中文：

```
language:
  label:
    fig: " 图 "
    tab: " 表 "
    thm: " 定理"
  ui:
    edit: " 编辑"
    chapter_name: [" 第 ", " 章"]
```


(4) 编译成书

Bookdown 可以编译成 HTML, WORD, PDF, Epub 等书籍格式。

点击右上窗口的 Build -> Build Book -> bookdown::pdf_book 启动编译, 其他可选还有 All Formats, bookdown::git_book, bookdown::epub_book.

若要导出到 Word, 需要在 _output.yml 增加:

```
bookdown::word_document2:  
  toc: true
```

则在 Build Book 下拉菜单出现 bookdown::word_document2 选项。

另外, bookdown 还提供了 publish_book() 函数让您很方便地将书籍发布到网上: <https://bookdown.org>⁴, Github 等进行分享。

⁴最新免费 R 语言书的大本营.

二. R 与 Git 版本控制

1. Git 版本控制

数据科学家通常是独立工作并与其他人共享随时间变化的文件、数据和代码。

版本控制是一个框架和过程，用于跟踪对文件、数据或代码所做的修改，其额外的好处是你可以恢复到以前的任何修改或时间点，您可以在同一个材料上与多人并行工作。

版本控制可以作为一种备份工作的方式，但真正发挥作用是用合作项目中。项目一般包含是由多人进行的一系列常见步骤，比如

- 下载/收集数据
- 清洗/变换数据
- 分析和可视化数据
- 生成精美的结果报告

这些任务包含相互重叠或相互依赖，复杂的合作项目需要预先考虑如何设置，以便每个人都能将自己的贡献无缝地衔接到项目的整体结构中，而不会耽误其他团队成员的进程。有了版本控制，如果您所做的修改破坏了一些东西，很容易通过提交时间线恢复或回到较早的工作版本。

Git 和 Github 是最常用的基于云服务的版本控制工具，能够准确控制哪些修改和/或获取文件“版本”。这些版本是您工作的快照，有唯一的标识符和简短的提交信息，您能够在任何时间点恢复这些更改。Git 对特定用户的特定修改提供了更精细的控制，使得版本控制成为一个非常强大的工具。

Git 和 Github 有什么区别？

- Git 是版本控制软件，安装在您的电脑上，有相关的命令，用来与版本控制的文件互动；
- Github 是与 Git 对接的网站，允许我们将文件作为仓库进行存储/访问/共享；同时 Github 也是全球最大的代码托管网站，里面有着世界各地程序员分享的海量程序代码。

注：在国内访问 Github 容易失败，可以换成 gitee (码云)，可称为 Github 的国内汉化版，所有操作几乎是一样的。

2. RStudio 与 Git/Github 交互

(1) 安装并配置 Git

到 Github 官网: <https://github.com> 注册账号。

到 Git 镜像站: <https://npm.taobao.org/mirrors/git-for-windows/> 下载对应系统版本的 Git 软件, 安装, 所有选项保持默认即可。

重启 RStudio 会自动检测并关联到 Git.

- 配置 Git (只需配置一次)

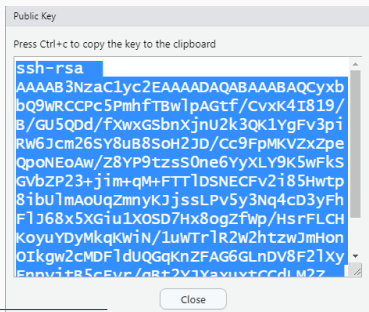
可以通过 Git Bash 操作, 更简单的方法是用 `usethis` 包:

```
library(usethis) # 建议用 Github 注册的用户名和 Email
use_git_config(user.name = "zhjx19",
               user.email = "zhjx_19@163.com")
```

- 用 SSH⁵连接 GitHub (只需配置一次)

用 SSH 密钥, 就不必在每次 RStudio 与 GitHub 交互时提供一遍用户名和密码。

RStudio 依次点开 Tools -> Global Options -> Git/SVN, 点击 Create RSA Key ..., 弹出窗口点击 Create, 完成后再点击 View Public key:



⁵SSH 协议, 连接和验证远程服务器和服务。


复制, 到 Github, 依次点击头像 —> Settings —> SSH and GPG keys —> New SSH key, 粘贴到 Key 框, 点 Add SSH key.

成功则显示:

SSH keys

[New SSH key](#)

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.


SSH

zhang@DESKTOP-807911J

SHA256:CM1QJo1AA1WcVsYS0wjIyvVqCaLSbvgUX1SOS/U6+rs

Added on 4 Aug 2021

Never used — Read/write

[Delete](#)

Check out our guide to [generating SSH keys](#) or troubleshoot [common SSH problems](#).

(2) 创建仓库

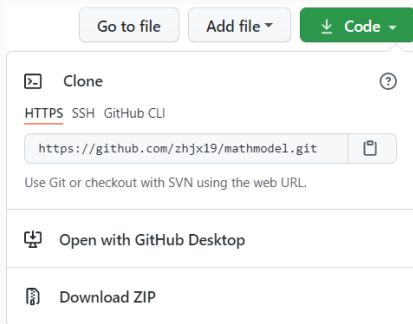
- 创建 Github 远程仓库

登录 Github 网站, 个人主页点击 Repositories -> New, 创建一个新仓库 (Repository), 比如起名为"test-demo", 选择 Public (公共仓库), 勾选下面的 Add a README file, 点击 Create repository, 则成功创建 test-demo 仓库。

注: 也可以 fork 别人的公共仓库到自己名下使用。

- 克隆仓库到本地

进入 Github 仓库页面，点击 Code 下来菜单，点复制按钮复制 HTTPS 下的仓库地址备用：



RStudio 依次点击 `Creat Project -> Version Control -> Git`, 在 `Repository URL` 框粘贴前面复制的仓库地址, 然后浏览选择本地路径, 勾选 `Open in new session`

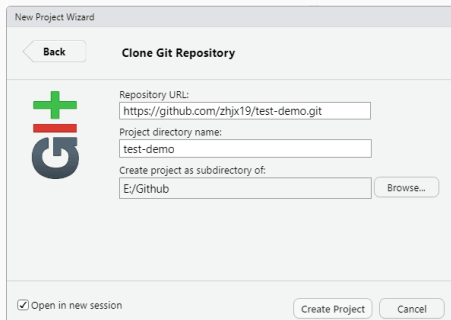


图 1: 新建带 Git 版本控制的 R 项目

点击 Create Project, 则开始从远程克隆到本地仓库, 等待完成。现在已有一个用 Git 控制的 RStudio 项目, 并且在电脑上有一个可以使用的本地副本。

可以查看和克隆任何公共 Github 仓库。然而, 只有所有者或合作者才可以提交修改。

- 添加合作者: 在 Github 仓库页面, 点击 Settings -> Manage access, 再点击 Invite Collaborator, 并输入合作者用户名或 Email, 搜索并加入。

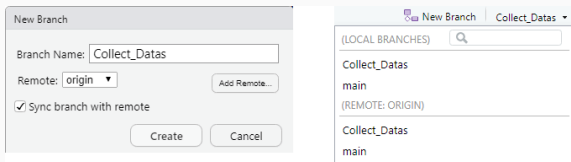
(3) 分支与合并

自己一个人就能负责的小项目，可以不用分支，都在主分支进行下文的一般操作流程：**暂存、提交、推送**即可。但是如果你想要实验某些新功能路线，没有问题再合并到主分支，或者你的项目需要多人分工合作共同完成，就非常需要做**分支与合并**。

通常需要确保项目的**主分支 (main)** 是干净的，并且是已接受的最新变化.....所有其他的工作将通过**分支 (branch)**、**拉回请求 (pull request)**、**审查 (review)** 和**合并 (merge)** 过程来减少冲突或重复的工作。

项目每个人是先从主分支，拉出一个分支做自己负责的部分，成功做完后再请求合并到主分支。

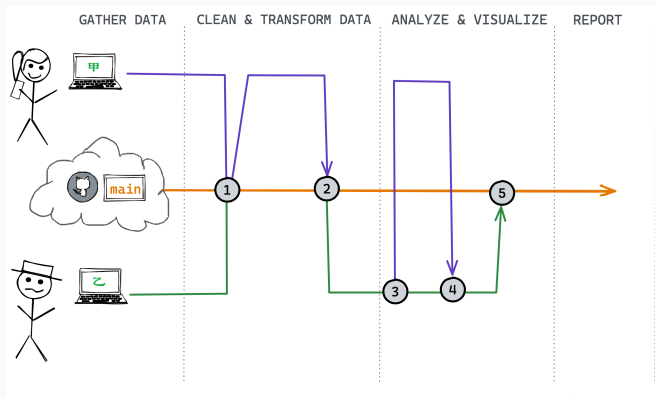
比如成员甲，在 RStudio 右上窗口切换到 Git，点击 New Branch 按钮，输入分支名比如“Collect_Datas”，确保远程设置为 origin：



然后点击 Create，则创建了一个名为“Collect_Datas”的新分支，可点开查看所有分支。

通常每个团队成员都分别创建自己的分支，这些分支都来自主分支。

负责分支项目的成员的一般工作流程示意图：



(1) 暂存 (stage)

做一些工作：文件新建/修改/删除。为了对这些变化进行版本控制，需要对这些工作进行**暂存 (stage)**，以便它能被 Git 版本化。

例如，创建 `datas` 文件夹，放入 5 个学生成绩的 `xlsx` 文件，新建 R 脚本 `read_datas.R`，内容是批量读取这些数据文件。

所有新增加或修改过的文件，都会在 Git 窗口显示出来，添加之前是黄色按钮，勾选 `Staged` 下面的选框，变成绿色则完成添加：

| Staged | Status | Path |
|--------------------------|--------|-----------------|
| <input type="checkbox"/> | ? | .gitignore |
| <input type="checkbox"/> | ? | datas/ |
| <input type="checkbox"/> | ? | read_datas.R |
| <input type="checkbox"/> | ? | test-demo.Rproj |

| Staged | Status | Path |
|-------------------------------------|--------|--------------------|
| <input checked="" type="checkbox"/> | A | .gitignore |
| <input checked="" type="checkbox"/> | A | read_datas.R |
| <input checked="" type="checkbox"/> | A | test-demo.Rproj |
| <input checked="" type="checkbox"/> | A | datas/六1班学生成绩.xlsx |
| <input checked="" type="checkbox"/> | A | datas/六2班学生成绩.xlsx |
| <input checked="" type="checkbox"/> | A | datas/六3班学生成绩.xlsx |
| <input checked="" type="checkbox"/> | A | datas/六4班学生成绩.xlsx |
| <input checked="" type="checkbox"/> | A | datas/六5班学生成绩.xlsx |

图 2: Staged 步：暂存修改

(2) 提交 (commit)

填写简洁的描述性的提交信息，建议添加一些有用的信息，便于回看。

Git 窗口，点击 Commit，弹出 Review Changes 窗口，在 Commit Message 窗口填写提交信息：

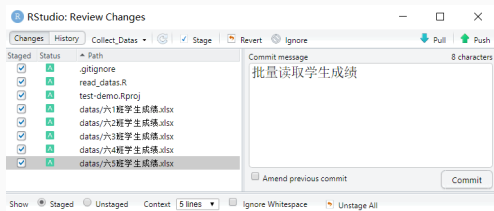


图 3: Commit 步：提交

点击 Commit 提交，等待完成。

(3) 提交 (push) 到“Collect_Datas” 分支

准备好本地提交的修改后，就可以将其推送到 Github 远程云仓库。记住，可以多次提交统一推送，或每次提交时推送，时间戳标识符是随着提交而不是推送信息添加的。

点击 push 按钮提交到“Collect_Datas” 分支，等待完成⁶。

提交成功后，Github 仓库页面，多了一条 Compare & pull request:

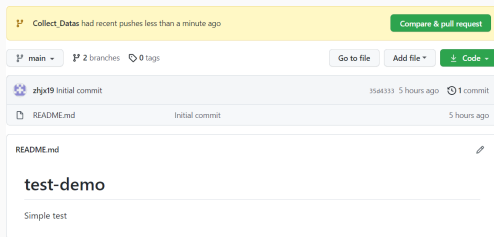


图 4: Push 步：提交并查看

⁶若 Github 网不通，则不能提交上去。

版本控制的下一步是将刚才项目成员的工作从“Collect_Datas”分支转移到主分支。这就用到“**拉取请求** (pull request)”。

拉回请求是一种在合并到仓库之前提出和讨论修改意见的工具，检查是否有冲突⁷，然后将这些修改合并到主分支。

然后，可以删除“Collect_Datas”分支，再创建一个分支来处理下一个任务，如此反复。

点开 Compare & pull request，可以填写一些额外的描述、评论，关于该拉回请求正在做什么，以及为什么。并为审查者（项目合作者）添加标签、重要事件等。这些都有助于跟踪哪些工作已经完成，哪些尚未完成。

⁷比如大家在同一个文件上工作。

然后点击 `Create pull request` 按钮，期望看到“This branch has no conflicts with the base branch（本分支与基础分支没有冲突）”，这意味着主分支可以很容易地将这项新工作合并进来！若有合并冲突则需要尝试解决它。

继续点击 `Merge pull request` 按钮，等待出现“Pull request was successfully merged and closed”（拉回请求被成功合并和关闭）！

注：拉取请求另一种常见用途是，分叉（fork）别人的仓库副本，对别人的代码做出改进，申请提交合并。

项目的下一项工作，比如探索性数据分析，由另一位项目成员乙负责，他首先需要将成员甲合并的改动拉回（pull）主仓库，保持更新到最新，有两种方法：

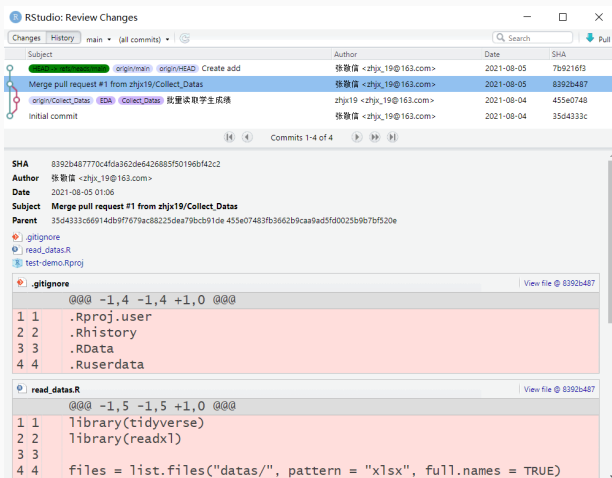
- `git fetch`: 安全做法，因为它从仓库中下载任何远程内容，但不更新本地仓库状态。它只是保留一份远程内容的副本，让任何当前的工作保持原样。为了完全整合新的内容，需要在 `fetch` 之后进行合并（merge）
- `pull`（拉回）⁸: 将下载远程内容，然后立即将内容与本地状态合并，但如果存在未完成的工作，这将产生**合并冲突**，但不用担心，这些可以被修复。

然后，如同成员甲一样，新建自己的分支，然后执行文件新建/修改/删除，再暂存、提交、推送、拉回请求、合并到主分支⁹。

⁸点击 Git 窗口的 `pull` 按钮。

⁹当然，成员甲如果需要对一些数据进行快速修改，也可以 `pull` 成员乙的分支，进行修改（提交），并在成员乙合并到主分支之前，将其合并到乙的分支。

Git 版本控制的上述操作流程最大的好处就是可以撤销错误，点击 Git 窗口的 History 按钮打开历史窗口：



列出了每个提交，点击查看具体提交内容，包括 SHA（唯一的 ID）、作者、日期、父级和提交的修改。

浏览找到发生错误之前的提交，记下 SHA, Git 窗口点击 More -> Shell 打开命令行窗口，执行：

```
git checkout <SHA> <filename>
```

就能回滚到错误发生之前的文件（注意是覆盖!）

另外，可以用 `usethis::edit_git_ignore()` 访问或创建 `.gitignore` 文件，其中包含的任何文件扩展名或特定文件都意味着 Git 会忽略它。因为有些文件，比如临时文件、日志文件、带有私人信息的 `.Rprofile` 文件，是不需要或不能向云端提交的。

本篇主要参阅 (张敬信, 2022), (Xie et al., 2021), (Xie, 2017), (李东风, 2020), (Hadley Wickham, 2021), 以及 R4WRDS: Version Control with git, Happy Git and GitHub for the useR, 模板感谢 (黄湘云, 2021), (谢益辉, 2021).

参考文献

Hadley Wickham, J. B. (2021). *R Packages*. O' Reilly, 2 edition.

Xie, Y. (2017). *bookdown: Authoring Books and Technical Documents with R Markdown*. Chapman and Hall/CRC. ISBN 9781138700109.

Xie, Y., Dervieux, C., and Riederer, E. (2021). *R Markdown Cookbook*. Chapman and Hall/CRC, Boca Raton, Florida. ISBN 9780367563837.

张敬信 (2022). *R 语言编程：基于 tidyverse*. 人民邮电出版社, 北京.

李东风 (2020). *R 语言教程*.

谢益辉 (2021). *rmarkdown: Dynamic Documents for R*.

黄湘云 (2021). *Github: R-Markdown-Template*.