

章节4 Git, GitHub and RMarkdown

4.1 实践目标 到本实践结束时，你应该能够：

1. 解释 Git 和 GitHub 的用途及差异；
2. 创建可重复和开源的 R 代码；
3. 生产用于解释代码和分析的 RMarkdown 文档。

4.2 课后作业

在我们安排的课程之外，你应该每周额外花大约 12 小时的学习时间。你可以自由选择你感兴趣的 GLS 主题，但以下是一些好的开始点：

每周测试 每周我们将提供一个小任务来测试你的知识，这些任务应该用于指导你的期末考试复习。

本周的任务是：

- 读取全球性别不平等数据
- 将全球性别不平等指数连接至世界空间数据，创建一个新的列来表示 2010 年和 2019 年间不平等的差距
- 将结果分享到 GitHub
- 在传播电子表格中添加你的仓库 URL

提示： countrycode R 包会很有帮助！

提示： 性别不平等在过去一年发生了变化，你可以在“所有综合指数和成分时间序列（1990-2021）”数据集中找到你需要的数据，该数据集下面的元数据文件将解释每一列的含义。

阅读材料

本周阅读：

- 第 2 章节 Xie、Allaire 和 Golemund (2019 年) 《RMarkdown：权威指南》
- 第 2 章节 “基础” Tierney (2020 年) 《科学家 RMarkdown》
- “为什么使用 RMarkdown” Goodchild 和 Wenwen (2020 年) 《社会科学和环境科学必须弱化时间和空间复制》
- Marwick、Boettiger 和 Mullen (2018 年) 《使用 R (及其朋友) 包装可复现数据分析工作》

观看材料

- Alice Bartlett (来自 UX Brighton) 的《Git 适合人类》
- Hadley Wickham 在欧洲分子生物学实验室 (EMBL) 的演讲。接下来几周都会观看这个演讲
- Karthik Ram 的《使用 R 进行现代可复现数据科学的指南》
- R Markdown 笔记本

记住这只是起点，探索阅读清单、实践和讲座以获得更多信息。

4.3 推荐音乐

其中部分实践可能会需要很长时间，请定期休息，并听听每周我们推荐的其中一首喜欢的歌曲。

Andy 美丽的人会毁掉你的生活！这是我最喜欢的乐队之一——**Wombats**。成立于 2003 年利物浦表演艺术学院。只有真正的才华横溢的音乐家才能做到这一点

Adam 当两位最伟大的说唱歌手拿起麦克风合作制作音乐时会发生什么？他们只会与顶级 **drum&bass** 制作人联手推出一张双专辑，绝对棒极了！是的，**Delroy** 和 **Dominic** 是他们妈妈眼里的名字，对于我们其他人来说就是 **DRS** 和 **Dynamite**，是《**Playing in the Dark**》！

4.4 引言

在本实践中，你将学习如何使用 **RStudio**、**RMarkdown**、**Git** 和 **GitHub** 产生可开放、可重复和可携带的工作。随着越来越多的研究者和组织将代码与其稿件一起发布，熟练掌握这些工具变得越来越重要。

你将使用的工具有：

- **RStudio** 是一个图形界面（你应该已经熟悉）——它具有一些功能，使其非常适合编写可复现和开源地理数据科学作品。
- **RMarkdown** 是一种 **Markdown** 格式化语言的一个版本，它可以使纯文本包含指向数据的链接、运行的代码以及解释你正在生产的内容和元数据，告诉你的软件从 **Markdown** 代码生成哪种类型的输出。有关 **RMarkdown** 的更多信息，请参阅此处。
- **Git** 是一个软件版本控制系统，它使您能够跟踪您产生的代码以及您或其他人对它的更改。
- **GitHub** 是一个在线仓库，可以让任何人查看您生产的代码（无论你选择何种语言）并对其使用、审查、贡献和评论。

4.5 Git和GitHub

4.5.1 三种方法

有三种方法可以让您的**RStudio**项目与**GitHub**一起工作。你将使用的工具有：

- 设置**GitHub**存储库，将其克隆到**Git**中，然后使用**Git GUI**加载到**RStudio**中。
- 创建新的**RStudio**项目并将其链接到**GitHub**-新的版本控制。
- 如果您有一个现有的**RProject**，则可以手动将它们链接起来-现有项目将显示所有三种方法，您应该能够以第一种方式执行，然后使用相同的存储库以第二种方式执行。第3种方法将遇到合并问题，因此请从一个新的**GitHub**存储库开始。如果您已经生成了一些代码，那么以后想要共享它时很有用。遵循我在讲座中的做法。

我的建议是在开始之前先阅读实践中的**Git**和**GitHub**部分（直到**RMarkdown**部分）。

4.5.2 设置您的GitHub

- 如果您在自己的计算机上工作，则首先需要安装**Git**--<https://git-scm.com/> -如果在UCL远程桌面 上工作，则不需要这样做，因为它已为您安装。
- 前往<http://github.com>，创建一个帐户并创建一个新的存储库（您可以随意命名，例如“**qis_code**”或其他类似名称），确保它是公开的，并选中“初始化新存储库的复选框带有**README**”，然后点击底部的“创建存储库”。

老板

库名称

安德鲁·麦克拉克兰

/ 例子

优秀的存储库名称简短且令人难忘。需要灵感吗?那迷你椰浆麦片怎么样?

描述(可选)

公共

任何人都可以看到这个存储库。你选择谁能承诺。

私人

您可以选择谁可以查看并提交到这个存储库。

如果导入的是现有的存储库，则跳过此步骤。

☒ 使用README初始化此存储库

这将允许您立即将存储库克隆到您的计算机上。

添加.gitignore:无

添加license:无

创建存储库

1. 你已经创建了一个新的仓库（repo）。你可以在这里存储你的代码。你会发现这里也已经创建了一个README .md 文件，这个文件可以编辑用来告诉人们在这个仓库里可能会找到什么内容。

4.5.3 使用RStudio配合Git

在夏季的2021年，GitHub改变了它的认证过程，从基于密码的方式变为基于令牌的方式。[David Keys](#)提供了一篇很好的概述，并包含一些视频来记录这个变化以及如何进行设置，我已经在此处进行了改编。

4.5.3.1 检查Git是否安装

在控制台窗口中，你会看到一个终端标签页。使用`which git`然后`git --version`检查Git是否已安装。你应该会收到一条消息回复，告诉你Git安装的位置和版本号。

4.5.3.2 配置你的Git

你需要告诉Git你是谁以及你的GitHub用户名。最简单的方法是使用`usethis`包，需要先安装并加载它。然后，在控制台中，输入函数`edit_git_config()`。
Git配置将会加载，你需要将名字和邮箱更改为与GitHub匹配。
如果这是空的，请使用以下模板并保存文件。

Code

4.5.3.3 启动Git

要启动Git，您需要在**一个RStudio项目**中。下面的指示教你如何根据不同的场景完成此操作。例如，在第一个Git方式中，我们将克隆（复制）远程仓库（我们自己的），如果我们想要更改（到远程），我们需要按照这些指示链接我们的Git到GitHub。

首先，我们只是从远程（GitHub）复制它，这将准备好git - 所以现在不要这样做.....但非常简单的方法是在控制台再次加载usethis包和use_git()函数，然后输入选项1。你会为此目的而做的是一个你已经开始的项目，但是没有启用git。

4.5.3.4 连接Git到GitHub

一旦我们在RStudio中有了一个带有Git的项目，无论是自己制作还是从GitHub下载，我们都需要将其连接到GitHub。

从GitHub开始，您需要生成一个个人访问令牌。您可以使用usethis包中的create_github_token()函数，也可以通过GitHub>设置>开发者设置>个人访问令牌>生成新令牌来执行此操作。

使用描述性的名称，并考虑保存令牌-它不会保存在GitHub上 最后一步是使用gitcreds包将此令牌存储在Git中>安装并加载它>然后使用gitcreds_set()函数>将您的令牌复制进去。

4.5.4 方式一 —— 使用Git GUI

- 1、现在你已经在线创建了你的仓库，你需要对其进行“克隆”，以便在计算机的本地文件夹中有其相同的副本。有几种方法可以实现这一点，但最简单的一种是使用与您的git安装一起打包的GUI。
- 2、首先要做的是从github网站复制你的仓库的Clone URL——点击你的仓库中的绿色按钮，“Clone或Download”，并复制链接：



1. 然后在Windows开始菜单中，转到Git > GUI。
2. 选择“克隆现有仓库”，并将GitHub账户中的链接粘贴到上方框中，并将本地目录粘贴到下方框中，用于存储你的仓库（注意，你需要为新建文件夹命名，一旦选择了现有目录，在Windows资源管理器中不要新建文件夹，必须在其文件路径中指定）。



1. 在几分钟之后，你应该可以在你的本地机器上查看GitHub仓库的副本了。这是你将要存储所有代码和一些其他文件的地方，以供你的可复现研究使用。
2. 打开RStudio并选择“文件” > “新建项目” > “现有目录”。



1. 将项目的 工作目录 设置为你在Git GUI目标目录中指定的内容。现在，你已经将你的项目链接到了你的本地Git。

请注意，当我们尝试从RStudio推送到GitHub时，按钮可能会被弹出，这很可能是由于你的本地Git分支没有跟踪(跟随)GitHub分支!我会在灰色按钮部分告诉你如何修复这个问题。

4.5.5 在RStudio中创建一个新的版本控制——方法2

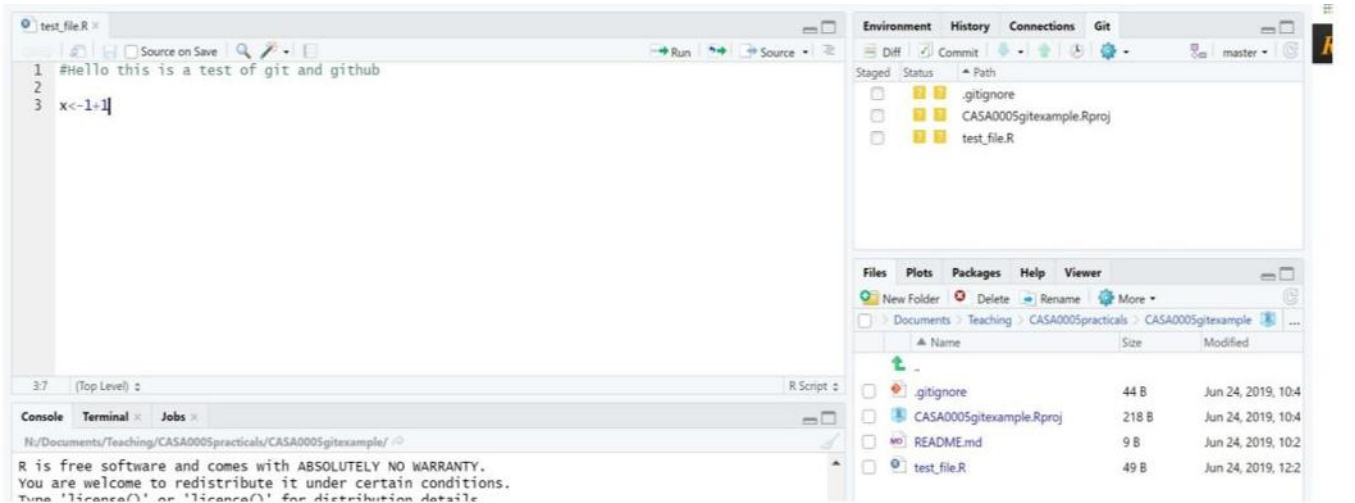
有一种更简单的方法可以设置与你的项目相关的Git和GitHub，但假设你是从头开始的（没有任何代码在RProject中）！

1. 在"设置你的GitHub"下，我们在GitHub上制作了一个存储库。复制那个URL。
2. 打开RStudio>文件>新建项目>版本控制>Git
3. 复制存储库URL，并提供项目目录名称.....但是当你粘贴URL时它应该会自动填充。



4.5.6 如果已有现有项目-方法3

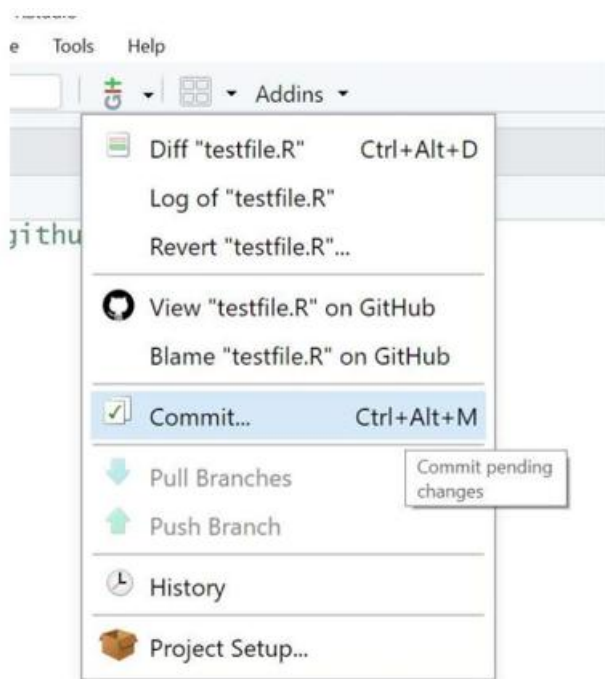
1. 在RStudio中打开现有的项目（或创建一个新的项目.....我会在这里创建一个）。在RStudio工具>全局选项下，选中Git / SVN下的允许版本控制框，并找到计算机上的git.exe文件所在的文件夹——如果您已安装git，那么这应该是自动存在的。如果创建新项目，请确保创建一个文件(.R或.Rmd通过文件>新建文件)，添加内容，然后将其保存到项目文件夹中(文件>另存为)。当保存时，它应出现在右下方文件窗口中。
2. 接下来转到工具>项目选项>Git / SVN>并将版本控制系统设置为Git。您现在应在RStudio环境窗口的顶部看到一个Git标签，文件也显示在Git标签下。看起来应该像这样.....



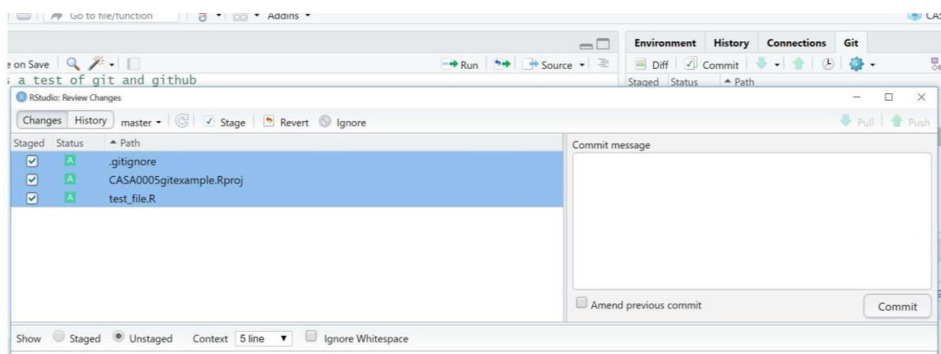
现在您可以按照以下指示使用Git和GitHub了

4.5.7 提交至Git

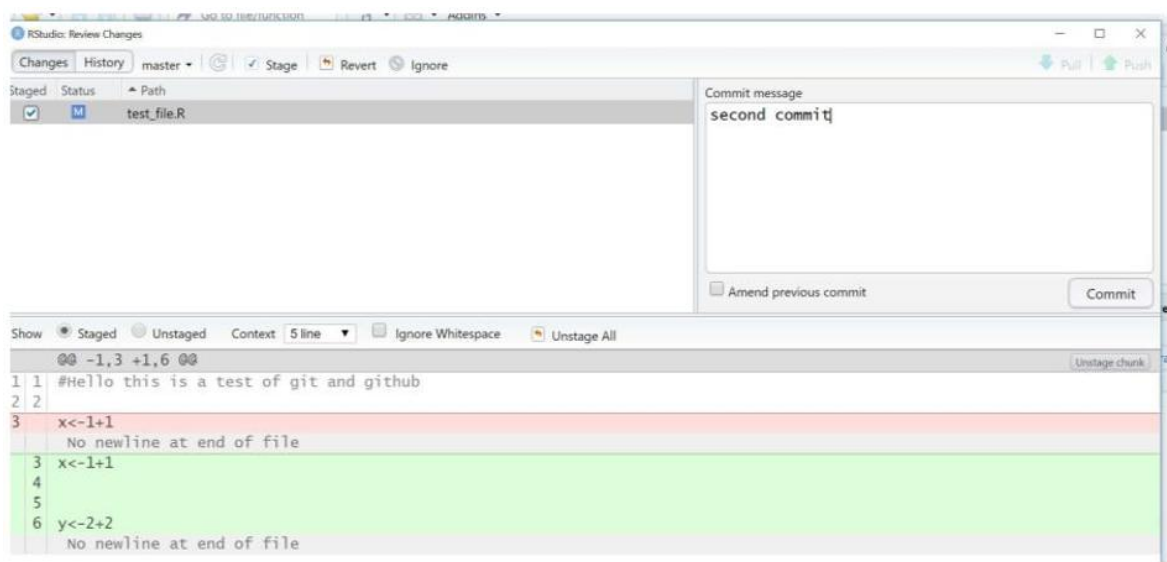
1. 除了保存（就像您通常所做的那样保存任何文件），我们还将“提交”或为我们的工作创建一个保存点git。
2. 为此，您应该单击“Git”图标，弹出类似下面的菜单：



你也可以点击出现在RS Studio右上角窗口的Git选项卡。Up会弹出另一个窗口，看起来有点像下面的窗口：



1. 分阶段更改，添加一个提交消息以便您可以监视所做的更改，然后单击提交。
2. 对您的文件进行更多更改并保存，再次单击提交，然后在“审阅更改”框中，您可以看到文件中发生了什么变化。添加一个提交消息，然后单击提交：



4.5.8 推送至Github

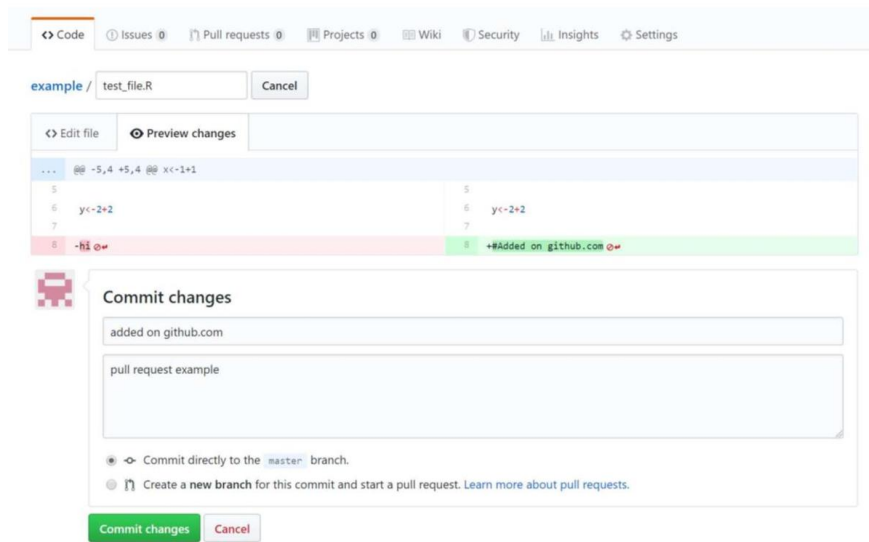
我们需要为我们的本地项目创建一个新的GitHub仓库。幸运的是usepackage可以为我们做到这一点。只需在控制台上输入函数use-github(), 就会出现一个使用您的项目名称的新GitHub仓库！

现在我们可以使用箭头向上按钮将更改推送到GitHub（位于RStudio Git选项卡环境象限）或从审阅更改框中推送（点击提交时打开）。

但是，如果推送按钮变灰，请转到灰色的推送按钮部分。

4.5.9 拉取自Github

1. 拉取会获取全球库中的任何更改，并将其引入本地库。转到示例GitHub仓库（在线）并单击测试文件>编辑此文件。
2. 添加一行代码或注释，预览更改，然后直接提交到主分支。



1. 现在在RStudio中单击向下箭头（拉取请求），您的文件应该会在RStudio中更新。如果您分别在GitHub和RStudio 中更新您的文件，当您尝试提交时，您将在RStudio中收到错误消息。

4.5.10 在RStudio之外使用Git

有时RStudio Git可能会有些麻烦。例如，在分段文件时，它们可能会花一些时间来显示带有勾选框的文件（我认为这是因为我们正在从网络中工作）。通常在RStudio中，您单击提交按钮，选择分段所有文件，等待几秒钟，然后关闭审阅更改框并从环境象限中的Git选项卡中的按钮提交。

另外，如果你想使用Git但你在UCL远程桌面或您在获得Git工作中遇到其他问题，不要害怕，你可以只使用原始的Git安装。

1. 在开始菜单中，打开Git GUI。开始>Git>Git GUI。你应该打开你刚刚创建的现有存储库。
2. 当您克隆存储库中的文件进行了更改时，可以使用git来检查更改，并将其“Commit”（保存）然后“推送”到GitHub的主要存储库中。
3. 若要检查并提交更改，在提交菜单中，只需：
 - a. 扫描更改
 - b. 阶段它们准备提交
 - c. 提交更改
 - d. 将更改推送到您的GitHub存储库

4.5.11 故障排除

4.5.11.1 是否挑战了您的密码？

自2019年1月以来，Git可能会使用操作系统提供的凭据助手。然而，截至2021年夏天，令牌系统已经取代了这一点，因此这种情况非常罕见。但是，您可以使用git提示手动设置用户名和电子邮件。转到工具>外壳程序并输入：

Code

这些只需要设置一次。

4.5.11.2 变灰的推送按钮

您的推送按钮是否变灰？当我尝试在[如果有现有项目]部分设置时，我的推送按钮是灰色的...别担心。

首先，让我们检查您的本地存储库(Git)是否连接到远程存储库(GitHub)。再次打开Shell(工具>外壳程序)并输入：

Code

这将显示您的fetch和push应该是GitHub上的存储库。如果需要设置远程存储库，请使用：

Code

用你的账户和仓库替换myname和myrepo——这就是我们从GitHub克隆的同一URL。

是否正确设置了？是的。

然后检查RStudio（和Git）当前分支是否跟踪远程仓库中的分支——我的没有。

Code

Origin/main表明本地main正在跟踪远程存储库中的origin/main。如果没有看到origin/main，则使用以下代码进行设置。目前，RStudio和git仍默认为主分支，所以第一行将更改为main，这需要与远程（GitHub）保持一致。

Code

```
Origin是您克隆的存储库(从GitHub)， main是分支的名称。你可能会看到类似…你的分支比origin/main早了1个提交。这意味着你已经提交了一些你正在本地仓库(Git)中工作的东西，这些东西还没有被推送到GitHub(源)和主分支…GitHub默认第一个分支名为main
```

如果你需要更改你的GitHub网址，也就是将本地Git推送到GitHub 账号（更改这一点），也许你已经建立了新的GitHub存储库。

Code

有关Git和GitHub的更多信息，请查阅 《Happy Git和GitHub for useR》 这本书。

4.5.12 创建副本

Github上的“Fork”是其他人存储库的一个副本，将其复制到自己的GitHub帐户上。您可以将其作为项目的基础起点，也可以提交修复，然后将其提交给原始拥有者，该拥有者会将您的更改拉入其存储库。

- 1. 可以从https://github.com/octocat/Spoon-Knife fork Github示例存储库
- 一旦fork成功，你应该会在自己的存储库中看到它

4.5.13 分支

在git中制作的每个存储库都有一个默认分支，但可以建立新分支来隔离特定工作的开发而不影响其他分支——就像测试环境一样。

- 1. 前往您刚才在Github上fork的test存储库。单击分支下拉列表，然后键入新分支的名称。



- 1. 现在单击README.md文件>编辑此文件
- 2. 添加一些更改，在屏幕底部完成更改框

嗯你好!

此存储库旨在提供在GitHub上分叉存储库的示例。

创建一个分支就是生成别人项目的个人副本。分叉在原始存储库和你的个人副本之间起着桥梁的作用。你可以提交拉取请求，通过提供你对原始项目的修改来帮助其他人的项目变得更好。分叉是GitHub社交编码的核心。

在分叉这个存储库之后，你可以对项目做一些改变，并提交一个Pull Request作为实践。

有关如何分叉存储库的更多信息，请查看我们的指南“分叉项目”。谢谢!



这是一个分支测试

$x < 1 + 1$



提交修改

更新README.md

分支示例

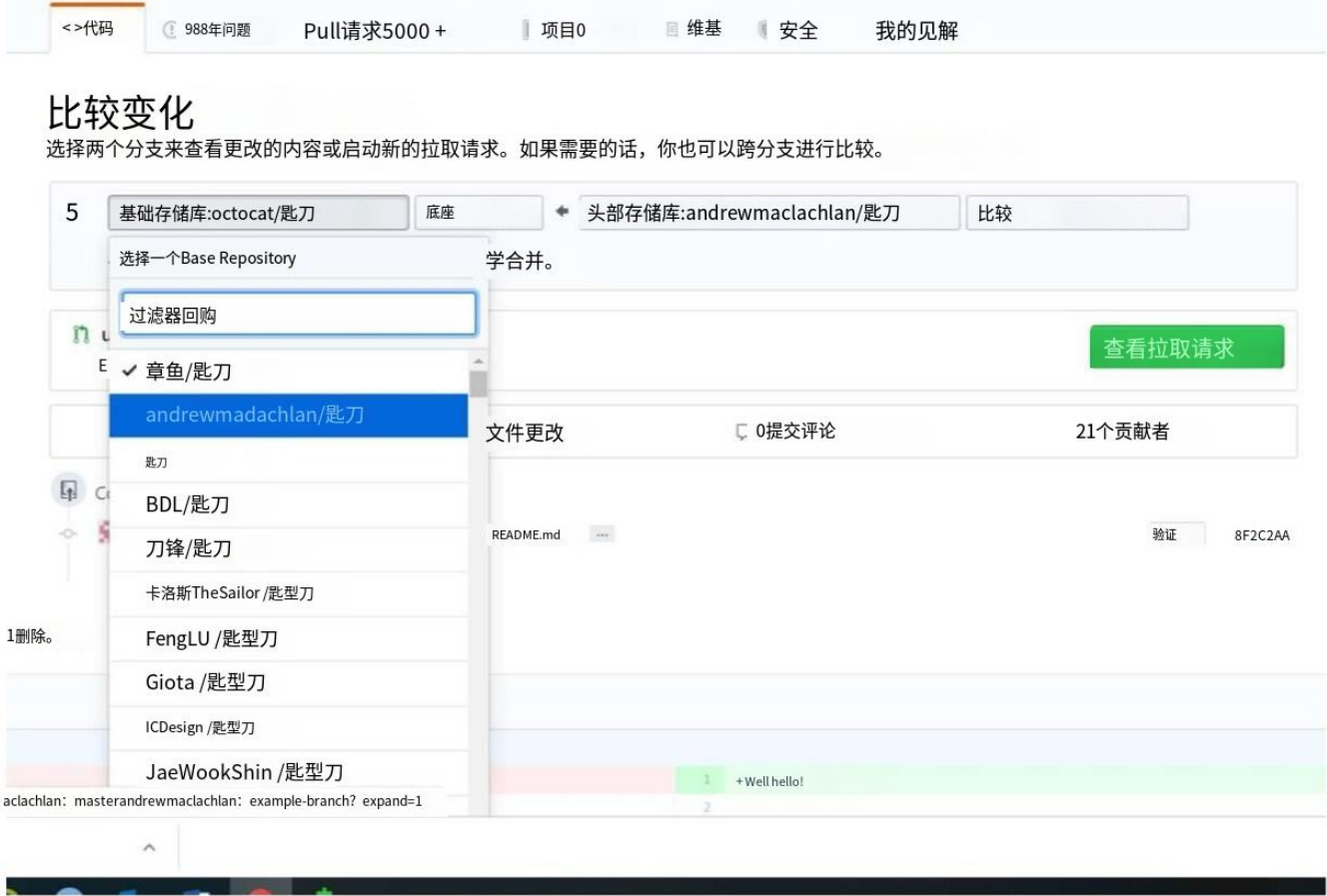
- ☒ < 直接提交到example-branch分支。
- ☐ n 为这个提交创建一个新的分支，并启动一个pull请求。了解更多关于拉取请求的信息。

1. 在这里，我们将直接向新分支提交更改。此时我们可以在此阶段对主分支做出更改，然后为它们创建一个新分支。提交更改。

2. 转到示例分支的主页（点击分支箭头并选择示例分支）。您会发现我们的示例分支现在比主分支提前了一个commit。

现在让我们创建一个pull请求到main分支。如果您修改了其他人的代码，那么您可以发送请求让他们拉入更改，此处是我们为自己发出pull请求，从我们的示例分支到我们的主分支。

1. 单击New pull request。
2. 在顶部将比较基础改为yours。



1. 现在滚动到底部，您会看到两个分支之间的对比。单击create pull request。
2. 选择squash并合并>确认squash和合并。这意味着我们在示例分支上的所有提交都被压缩成一个，因为我们只有一个分支，但是在将来可能会有用。
3. 返回主分支存储库，您应该会看到来自示例分支的更改已被合并。我们将稍后向您展示如何在网上发布RMarkdown文档。

4.5.13.1 Git命令

如果你想使用shell来控制Git，你可以这样做。如果您的项目很大，那么RStudio有时候会对文件名长度有所限制（比如书籍）。要绕过这个问题，可以使用以下命令：

- git add . to stage all files
- git commit -m "commit comment" to commit all the staged files
- git push to push the committed files to the remote

4.5.14 健康警告

在提交、推送和拉动时要注意避免合并冲突。每次都要思考一下自己正在做什么。GitHub帮助页面相当全面。<https://help.github.com/en/articles/resolving-a-merge-conflict-on-github>

4.6 RMarkdown

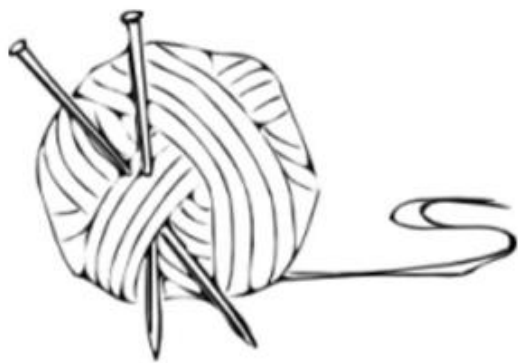
好了，现在已经完成了设置，可以成为一个可重现研究者！接下来就是进行可重复的研究。

R Markdown很棒，因为它可以在同一份文档中显示代码、解释和结果！由于方法论/用户指南中的信息不足或者步骤不符合最新版本的软件，因此常常难以重现结果。回想一下您不得不使用软件并参考大量用户指南的时刻，对我来说是一次痛苦的经历。R Markdown是一个很大的改进，它可以将所有信息放在同一个文档中，然后转换为多种不同格式——HTML网页、Word文档、PDF、博客、书籍等等。



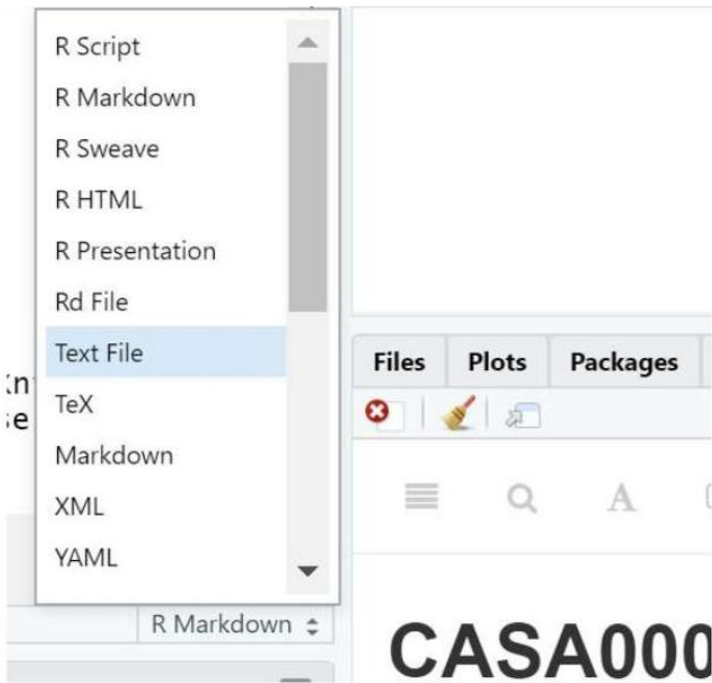
还可以不局限于R语言代码！要改变代码段之间的`{}`，只需编辑代码段内的Python代码即可，在R中默认情况下，您会得到`{r}`，只需将其更改为`{python}`。你还需要安装Python软件和R的`reticulate()`包。

在这次练习中早些时候，我要求您打开一个新的R脚本。现在你需要打开一个新RMarkdown文档。笔记本最初可以让您运行独立执行的代码块，但如果你选择Markdown文件也可以实现这一点。就我所知，唯一的区别是R Notebook会在头部增加输出：`html_notebook`文件的输出选项，在工具栏上添加预览按钮。如果没有此功能，则预览选项将被替换为Knit

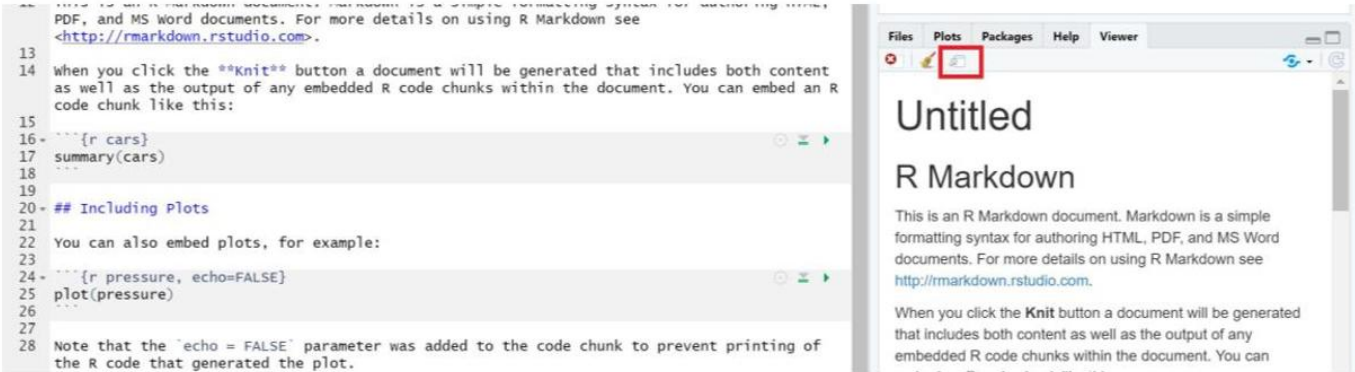


但是如果你希望得到预览按钮的话可以在头部混合输出选项。基本上没有太大的差别，可以通过一行代码手动更改它，可以查看stack Overflow上更多的信息。为了方便起见，最好坚持使用R Markdown文件有两种创建R Markdown文档的方法：

- 1.File>新建> R Markdown
- 2. 你可以在脚本窗口右下角更改类型

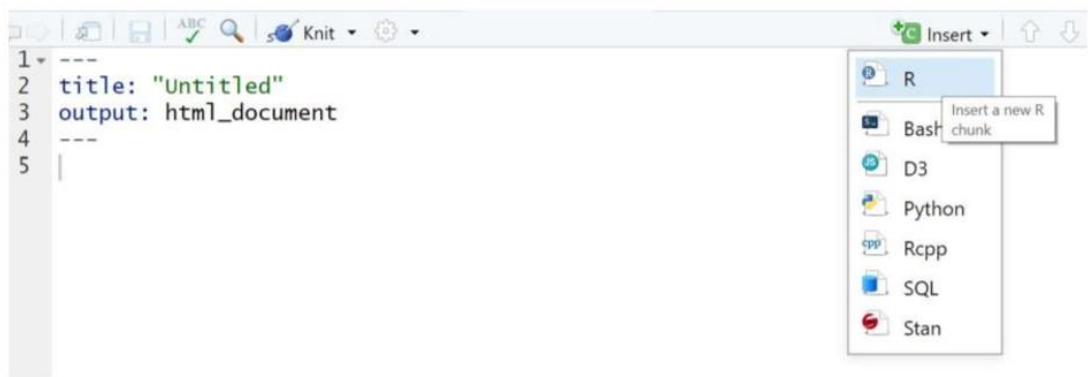


我一直使用第一种方法（所以在这里也使用这种方法），然后将会有一些示例数据，单击knit查看结果，文件将会加载到查看器窗格中，如果单击箭头和浏览器按钮则会在Internet浏览器中打开。



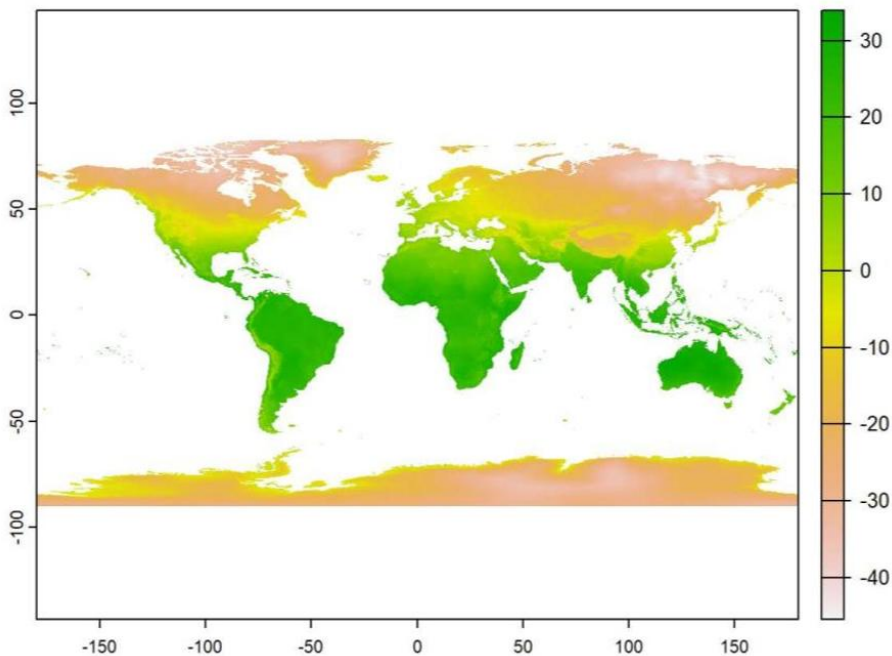
4.6.1 HTML

- 1. 现在将在新R Markdown文档中插入实际操作的代码...删除所有代码，只留下-
- 2. 在R Studio中可以选择Code > 插入代码块或单击 “Insert”按钮并将R代码插入其中



- 1. 出现一个小框，在这个框中您可以输入和运行R代码，并尝试粘贴以下内容:

Code



在您的作品中包含代码块时有各种选项允许您执行诸如包含代码而不运行它、显示输出但不显示代码、隐藏警告等功能。这些都可以通过单击代码块右上角的小齿轮图标自动输入，也可以在代码块标题中指定，切换按钮将会看到代码块标题中的代码变化有两个有用的图标位于设置小齿轮右边，第一个运行当前上方的所有代码（向下播放符号），第二个运行当前代码块（常规播放符号）



4.6.2 Knit选项

其他选项和技巧可以在这里找到完整的 R Markdown 指南：

- <https://rmarkdown.rstudio.com/lesson-1.html>
- <https://rmarkdown.rstudio.com/lesson-3.html> (for code chunk options)

4.6.3 快捷键

我找到最好的资源之一是由We are R-Ladies发起的一条[Twitter线程](#)以下是常用快捷键：

新建代码段CTRL+ALT+i

新增注释CTRL+SHIFT+C

对齐代码：CTRL+i

整理丑陋的代码使它看起来很好看：CTRL+ALT+A

插入折叠式导航标签 - 只适用于.r文件而不是.Rmd文件但仍很有用：CTRL+SHIFT+R

4.7 进一步阅读

自从开始编写这份指南以来，我发现了一本关于Jenny Bryan和Jim Hester编写的书Happy Git and GitHub for useR中使用R和GitHub的内容很棒，值得一看，还有一篇GitHub指南

4.8 反馈

本周是否有不清楚的地方？请告诉我们，请在反馈表单中匿名填写，我们将根据回复在未来改进材料。