

Reproducible Research using RMarkdown and Overleaf

Introduction

This is a tutorial for using Git to interface between Overleaf and RStudio, such that analyses conducted in an RMarkdown document can be incorporated into manuscripts and other writing in Overleaf. This could also be adapted for use in other LaTeX environments, but that is not the focus here.

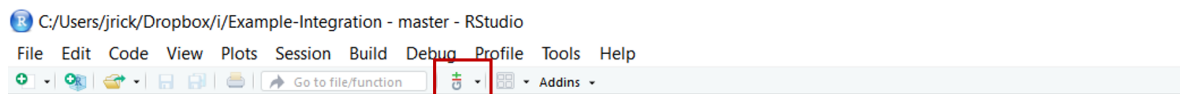
There are two main ways that this interfacing between RMarkdown and Overleaf can be used: (1) for including figures or tables created in R directly into Overleaf documents, and (2) for including code from R (and the results of that code) directly into an Overleaf document. The second requires a bit more work than the first, but both can be useful to ensure the transparency and reproducibility of research.

Tutorial

The first requirement to be able to do this is to have Git and RStudio installed on your computer in some manner. I would also recommend having a Github account, as I find this is the easiest way to make this whole process work. If you do not already have Git installed, there is a nice tutorial on how to do this [here](#).

*Important note for Windows users: if you download Git, it will automatically be installed into your **Program Files** directory. The space in this directory name will almost certainly cause you problems at some stage in the process, so I would recommend choosing a different location for installing, such as your `C:/Users/username/bin/` directory.*

Once you have Git installed, you need to make sure that your RStudio knows where to find it. Open up RStudio, and click on the **Tools** —> **Global Options** menu item. In the options screen, click on **Git/SVN**, and then enter or select the path for your Git executable and click “OK”. Once you open a R project, you should now see a little Git icon on the menu bar, as shown below. Now, you’re all set up and ready to start the conversation between RStudio and Overleaf.



In Overleaf

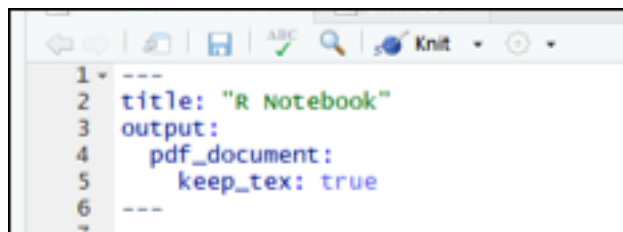
The first thing that you’ll want to do is set up an Overleaf project for the project that you want to work on. If you already have one, even better. With your project open, click on the **Menu** button in the upper lefthand corner. In this menu, you’ll see two options that we’re interested in: **Git** and **GitHub**. The **Git** option uses the Overleaf Git repository for the project, while the **GitHub** option creates a repository for the project in your GitHub account. Either option will work for what we’re doing, but my personal preference is to use GitHub, as I then have an automatic backup copy of my project should Overleaf start having

problems (note: this preference comes from personal experience, and the stress that comes from the Overleaf servers going down).

If you click on the **Git** option, a pop up will appear that gives you the address of the Git repository for your Overleaf project. If you're going with this option, you'll want to copy this/note it down, as we'll need this address once we get to working in R. If you go for the GitHub option, then clicking on the **GitHub** option will prompt you to connect your GitHub account to your Overleaf account (if you haven't already done this), and then to push your changes to GitHub. The GitHub repository will be the same name as your Overleaf project, so this is something to consider in naming your projects. Once you have your GitHub account linked, go ahead and push your project to the GitHub repository, and also make note of the GitHub address for your repository (it will generally be `https://github.com/USERNAME/PROJECTNAME.git`). Now, we're ready to move to R.

In RStudio

1. Open RStudio.
2. File – > New Project – > from version control – > Git
 - Here, you'll enter the repository URL from Overleaf. If you're just using the Overleaf Git repository, then you'll copy that address; otherwise, you'll copy the address from the GitHub repository that you created for your project. Do not change the project directory name.
 - Choose where you'd like the local version of this repository to be located. This will copy the git repository to your local folder, and you'll see the files in the Files tab inside of Rstudio
 - Check the box to start your project in a new session, and click **Create Project**.
3. Now, you'll see the files from your Overleaf project in the **Files** tab in RStudio. Thus far, these are likely all LaTeX-related files, and not anything to do with R. Now, open a new R Notebook.
 - You'll need to edit the YAML header of the R Notebook (the specifications at the very top of the document). Change the title of your document to something useful, and then replace `html_notebook` with `pdf_document`. On the next line, tab over and enter `keep_tex: true`, as shown below.



- Karl Broman has some really good tips for making your code and R Notebooks clean and usable, which I would recommend taking a look at if you're planning on integrating your code into your Overleaf documents: [see blog post here](#). One

particularly relevant tip is that you can provide a name for each of your code chunks in your R Notebook, which then becomes the name for the figures produced by those chunks. Another tip that I find particularly useful is specifying global knitr options for chunks at the beginning of your document, such as where figures should be written to (which is helpful for organization within your repository) and default figure sizes. The global options that I generally use look like this:

A screenshot of an R Notebook code chunk showing global knitr options. The code is as follows:

```
8- ##{r global_options, include=FALSE}  
9- knitr::opts_chunk$set(fig.width=12, fig.height=8, fig.path='figures/',  
10 echo=TRUE, warning=FALSE, message=FALSE)  
11 ##  
12
```

4. Now, fill out your R Notebook however you'd like, and knit it to a PDF. If you specified a `fig.path` in your global options, you'll notice that you now have a `figures` directory in your repository, which contains all of the output figures you produced. You'll also have a `.tex` document with the same name as your R Notebook file.
5. Once you have these things, it's time to push the changes back to Overleaf. To do this, you can either use the Git button on the RStudio menu, or use the RStudio Terminal (which should be open as a tab next to the Console). If your terminal isn't already open, you can open it from the menu through **Tools** — **Terminal** — **New Terminal** or through opening the shell (**Tools** — **Shell**). If using the terminal or shell, you'll want to type these commands:

```
git add . # to add all new files; otherwise, specify which files to add  
git commit -m "message"  
git push -u origin master
```

Note: if your terminal doesn't recognize the command "git", you may need to either add it to your path, or add an alias that directs it to your git installation.

Now, your R Notebook and associated files should be updated to your GitHub repository, or directly to the Overleaf Git repository (if you opted not to use GitHub)!

6. **IMPORTANT NOTE:** If you have already created your project and just want to pull most updated version, you'll use `git pull origin master`. It is always a good idea to make sure to pull the most recent version before beginning any analyses or making any changes, so that you don't have to deal with conflicts later on.
7. If you're interested in incorporating your R Markdown text and code in your Overleaf document (rather than just using the figures produced), then you'll need to add an additional processing step following knitting your document and prior to pushing your changes. The `.tex` file for your R Notebook automatically includes header information, which is unnecessary if it's going to be incorporated into an existing `.tex` document. This unnecessary information can be trimmed from the document by using something like the simple `trimtex.sh` script, which can be found at <https://github.com/jessicarick/r-overleaf/blob/master/trimtex.sh>. As long as this script is in the same directory as your R Notebook `.tex` document, you can run `./trimtex.sh filename.tex`, and this should remove the header information. You can then push all of your changes to the Git/GitHub repository.

Back In Overleaf

If you are using the Overleaf Git repository, your pushed changes should automatically appear in your file directory on Overleaf. With GitHub, you will need to click on the **Menu** – > **GitHub** link again, and manually pull changes from GitHub. You should now see all of the files associated with your R Notebook in your Overleaf project, including the `.tex` document and `figures/` directory. Any figures can now be incorporated into your Overleaf documents, just as you would do with figures manually uploaded from your computer. If you have to go back and change analyses in RStudio, these figures will now be automatically updated when you push those changes to GitHub!

To incorporate your R Notebook `.tex` file into your Overleaf document, you'll need to add `\usepackage{latex-rmd}` to your list of included packages in your `main.tex` file. The code for this package can be found at <https://github.com/jessicarick/r-overleaf/blob/master/latex-rmd.sty>. Then, you can just add `\input{filename.tex}` wherever you would like to include this file (where `filename.tex` is the `.tex` file associated with your R Notebook). If you are using this method to include your code in your Overleaf project, please make sure to tidy up your code (e.g. suppressing warnings and messages) – again, there is some good advice on how best to do this [here](#).

Once you've made any edits in Overleaf, be sure to push them back to your GitHub repository so that everything stays up-to-date. If you need to make changes to your figures or analyses, you can then pull the changes into your R Project, make changes, knit the document, push them back to the repository, and pull them down into Overleaf.

Adapted from [this Medium tutorial](#)

Written 26 Sept 2018 by [JA Rick](#) for Overleaf V1

Updated 26 Feb 2019 by [JA Rick](#) for Overleaf V2

Please contact jrick@uwyo.edu with any concerns, edits, or general feedback.