How to use GitHub

Dominique Maucieri

21 January, 2022

Contents

Getting started with Git

What is Git?

Git is a version control software that uses local files allowing ease of merging and branching.

Github is an interface that very user-friendly, helps with use of git software and collaboration with others. This makes it a popular choice for open source projects.

The Basics

Working with .Rmd files

Markdown files allow for you to create neat outputs of your code and comments.

How to create and set up R Markdown files

- To create a new R Markdown, go to "File >"New File" > "R Markdown"
- In the popup, title your file and list yourself as the author. Here is where you specify the format of your output, .html or .pdf.
  - This choice is up to you .pdf may require additional programs on your computer, but it will create cleaner looking outputs, while .html can support some functions that .pdfs cannot such as interactive plots
- Once you have your .Rmd file created, edit the top information so it is informative and will set up your R Markdown file, with your name and title:

For an .html knit output: title: "First R Markdown file" author: "Dominique Maucieri" date: "21 January, 2022" output: html_document: toc: yes toc_depth: 3 toc_float: yes

For a .pdf knit output: title: “First R Markdown file” author: ‘Dominique Maucieri’ date: “21 January, 2022” output: pdf_document: toc: yes toc_depth: 3

- This information will create the title for your outputs and allow you to format the table of contents of your final knit output. The date argument will automatically save the date at the time of knit. Knitting will take your R Markdown file and turn it in to a more polished final document that can be sent to collaborators to illustrate and compile your findings.
- You can choose to knit your R Markdown into an .html or a .pdf file, I will be using both so you can compare the difference. In order to have a .pdf output you will need to have a tex editor on your computer. https://ciser.cornell.edu/rmarkdown-knit-to-html-word-pdf/ may help if you want to make a .pdf output. No additional programs are needed for a .html output.

Syntax

- The use of “#” will allow you to create table of contents. The toc_depth argument above will tell you how many levels you can have in your table of contents (I have specified 3). A single “#” will be the largest table of contents heading, “##” will be smaller and “###” will be even smaller. Headings will be nested within the larger ones. More “#” the smaller the heading.
- https://rstudio.com/wp-content/uploads/2016/03/rmarkdown-cheatsheet-2.0.pdf

There is a lot of syntax that can help you to make effects within your text.

You can **Bold** text with two “**” on both sides of the words you want to bold

You can *Italics* text with one “*” on both sides of the words you want to italicize

You can make lists with one “*“, then a”+” and even a “-”

- First list item “*”
  - list sub-item “+”
    - list sub-sub item “-”

How to code R in R Markdown

In R Markdown, the text here will be normal text in your final output. To add R code at the top right of the script select “Insert” > “R” then in the curly brackets {} it will say r, add a title for that chuck such as “setup” as I have in the chunk below (can only see if you open this code in RStudio)

{r setup}

As long as you are coding within the R chunk, coding will be exactly the same as a .R file, other than setting a working directory. You may need to set the working directory in each chunk where you are loading or saving data as this only sets the working directory in a single chunk. Or to get around this you can work with R Projects as we will discuss later.

To run your code, you can either run each line individually, or highlight select rows and use the same commands we used in the .R script files, or if you want to run a whole chunk, you can select the play button at the top right of the chunk. There are many different ways to run an R Markdown file, and you can see these options in the pull down menu at the top right of the script, when you click ‘Run’

To knit the data together, you use the knit function at the top left of the script. This will knit based on how you have specified your document above in ‘Setup’.

```
library(MASS)

rm(list = ls(all=T)) # this code will clear your environment, giving you a clean slate from
# which you can start your new script
```

```
dev.off() # this code will close any plots that are currently open
```

```
## null device
##           1
```

What you see when this is knit is a lot of output. This is what would be outputted in the console when the code is run. But sometimes we don't want this to be seen in our final knit. To remove output and other things, we can add arguments to the curly brackets where I named the chunk {r setup}

Syntax examples Notice how I used 4 "#" and this heading did not show up in the table of contents, as I specified the levels I have in my table of contents and I only chose 3.

- {r setup, echo = TRUE}
  - will keep code the code in your knit output **use this as we want to see your code in the output**
- {r setup, warning = TRUE}
  - will show warnings in your final knit output, if you don't want them to show you could use warning = FALSE instead
- {r setup, message = FALSE}
  - will remove any messages from your final knit output
- {r setup, results ='hide'}
  - will hide the outputs from the code, however this will also remove plot outputs
- {r setup, fig.height = 3, fig.width = 3}
  - will specify plot output in inches, this example is a figure that is 3x3 inches

```
dev.off() # this time there will be no output
```

Sometimes you want to show the dataframe and outputs in your final knit though

```
data(Animals)
Animals
```

```
##                       body  brain
## Mountain beaver      1.350    8.1
## Cow                465.000  423.0
## Grey wolf           36.330  119.5
## Goat                27.660  115.0
## Guinea pig           1.040    5.5
## Dipliodocus      11700.000   50.0
## Asian elephant    2547.000 4603.0
## Donkey             187.100  419.0
## Horse              521.000  655.0
## Potar monkey        10.000  115.0
## Cat                  3.300   25.6
## Giraffe            529.000  680.0
## Gorilla            207.000  406.0
## Human               62.000 1320.0
## African elephant  6654.000 5712.0
## Triceratops       9400.000   70.0
## Rhesus monkey        6.800  179.0
## Kangaroo            35.000   56.0
```

```
## Golden hamster       0.120    1.0
## Mouse                0.023    0.4
## Rabbit               2.500   12.1
## Sheep               55.500  175.0
## Jaguar             100.000  157.0
## Chimpanzee          52.160  440.0
## Rat                  0.280    1.9
## Brachiosaurus    87000.000  154.5
## Mole                 0.122    3.0
## Pig                192.000  180.0
```

```
str(Animals)
```

```
## 'data.frame':    28 obs. of  2 variables:
##  $ body : num  1.35 465 36.33 27.66 1.04 ...
##  $ brain: num  8.1 423 119.5 115 5.5 ...
```

For more help with R Markdown

https://ourcodingclub.github.io/tutorials/rmarkdown/

Creating R Project

1. click "Project: (None)" in the top right corner of a clean R session in RStudio
2. click "New Project . . . "
   (a) "New Directory" if you do not have an existing folder/repository with your data or scripts in
      - here you will be asked to name this folder (Directory name), choose the location for this folder on your computer (Create project as subdirectory of) and choose if you want to create this as a git repository
      - in-order for this git repository to be initialized, you need to specify this in the git terminal, and because of this I think it is more difficult and I don't recommend. Instead I would follow (b) or (c)

   (b) "Exiting Directory" if you already have a folder that you are working in, and want to turn it into a project.
      - here you will choose the location of the fold with "Project working directory"
      - this is a good choice if you already have your repository on your computer

   (c) "Version Control" if you have a repository in git, and you want to clone it to your computer and create a project
      - here you will be asked to enter the "Repository URL" found on the GitHub repository under the green "Code" button

      - it will enter the directory name automatically and you can choose where to save the repository on your computer with "Create project as subdirectory of:"
3. Once specifying how to create the project, it's all set up. Now when you want to open your files, just double click on the project and it will set up RStudio for you, and where you left off. You may need to commit and push to git if this repository is on GitHub.

Working with R Projects

- Checkout the simplified_filepaths.R file and then the simplified_filepaths.Rmd file for why these R Projects are so useful and should be considered when working with R Markdown files