Guide to translation of the Epi R Handbook

Neale Batra

First draft 7 August 2021

[**Get ready to start with RStudio, Git, and Github**](#_gnsru6d03yg3) **1**

[Get contributor access to Github repository](#_h8il13x11q9o) 1

[Clone the language repository](#_3ozxszksr3yw) 3

[Orient yourself to the repository files](#_anqeyvvqpdkx) 6

[Finding the correct file to work on](#_3gxjdlxb9vc5) 7

[**Prepare for translation**](#_478oldayk9p1) **10**

[Make a branch to not disrupt others](#_mbpgyklny2yb) 10

[**How to translate**](#_njtqopemisic) **13**

[Decide your approach for translation.](#_oev3xr1ivjgb) 14

[Edit within the R Markdown “Visual Mode” with the help of DeepL.com](#_tv3dgwdv8lny) 14

[Commit and Push your changes](#_92uox7ytgt5n) 22

[Submit a Pull Request](#_nwpeuxb8qdfu) 25

[**How to “Preview” the page**](#_a4aeie5k6f9n) **28**

For other resources, see the Epi R Handbook page on [Version Control and Git and Github](https://epirhandbook.com/version-control-and-collaboration-with-git-and-github.html).

## 

## 

## **Get ready to start with RStudio, Git, and Github**

### **Get contributor access to Github repository**

1. **Install Git**

Download the Git software from [this website](https://git-scm.com/downloads). It will link to your RStudio.

1. **Establish a free account at Github.com**
2. **Install Github Desktop**

Not required, but it will be helpful. Install [from here.](https://desktop.github.com/)

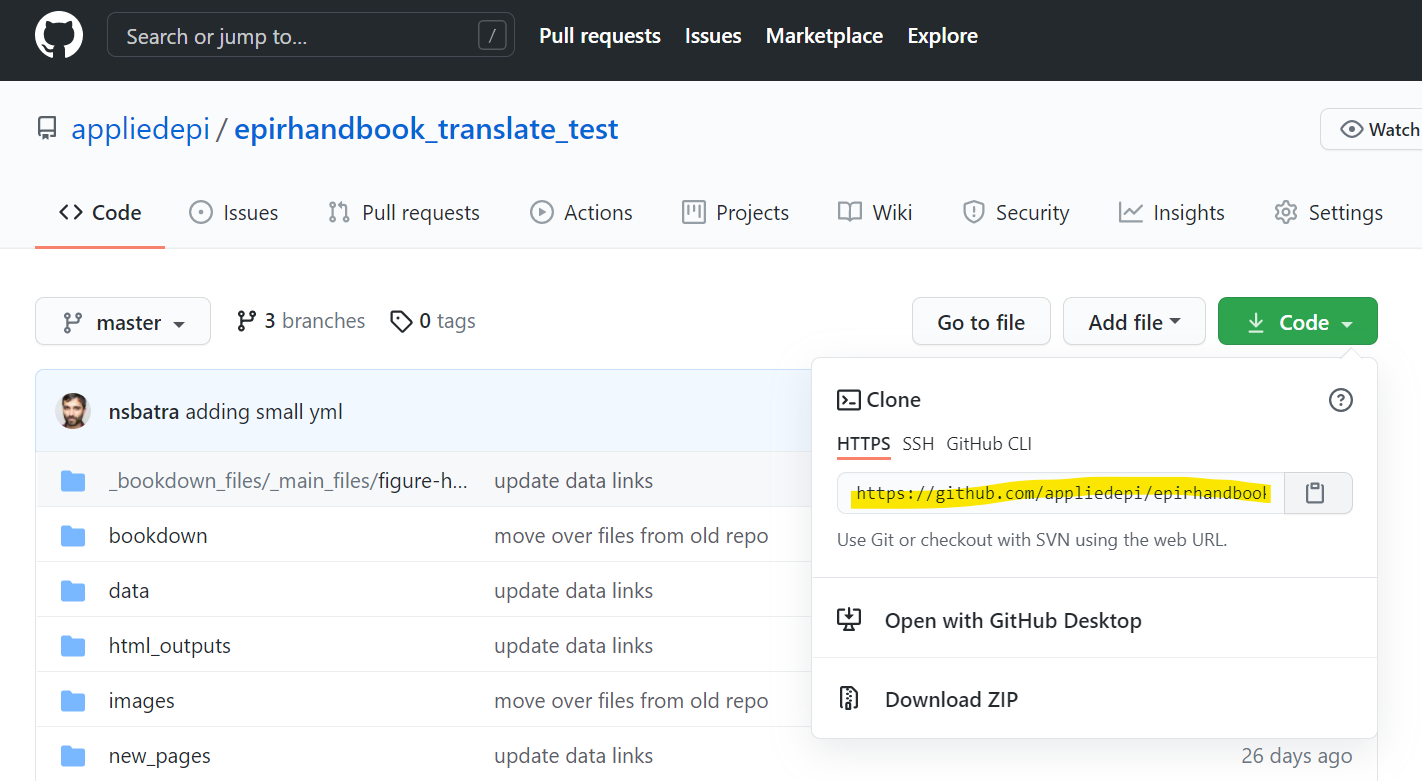
1. **Send your Github ID name** to Neale Batra (neale.batra@gmail.com), or to Alex Spina ([alexfspina@gmail.com](mailto:alexfspina@gmail.com)) if Neale is unavailable. For example, the Github name is “nsbatra” for Neale.

Once your ID is added as a contributor to the repository, you will receive an email. Please confirm acceptance. Then you will be listed as a contributor in the repository.

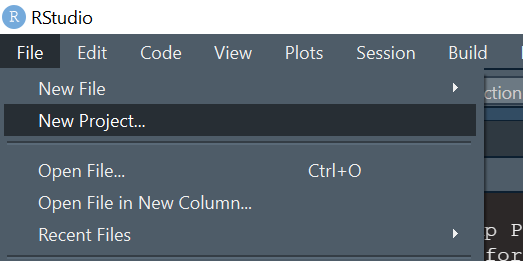
Your name should be listed below, in the “Settings” / “Manage access” of the specific repository

#### **Clone the language repository**

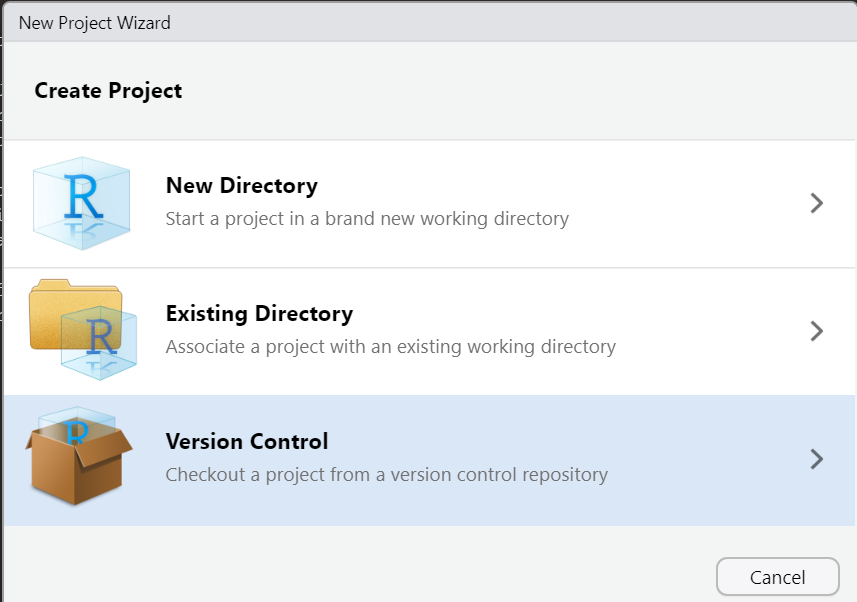
1. **In the Github website, open the repository of the language you are translating**. For example, for French the repository is “epirhandbook\_fr” and for Vietnamese it is “epirhandbook\_vn”.
2. **Click the green “Code” button and copy the HTTPS key**. This link should end with “.git”.



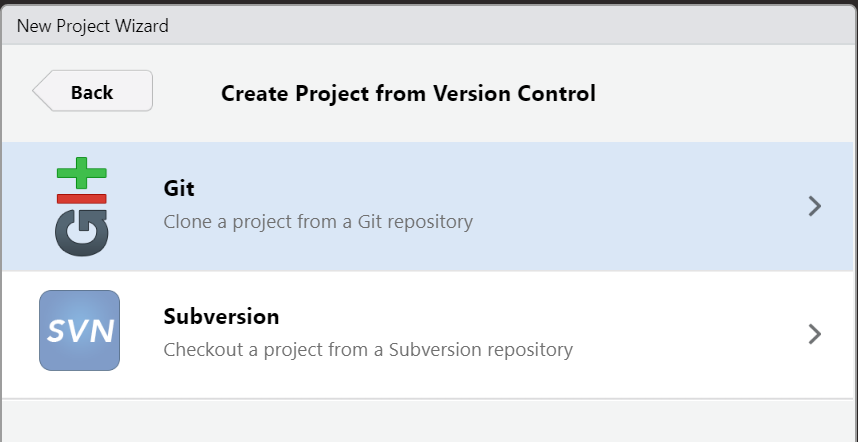
1. **In your local RStudio, open a new R project** using “Version Control”. You will link this R project to the Github repository for your language.



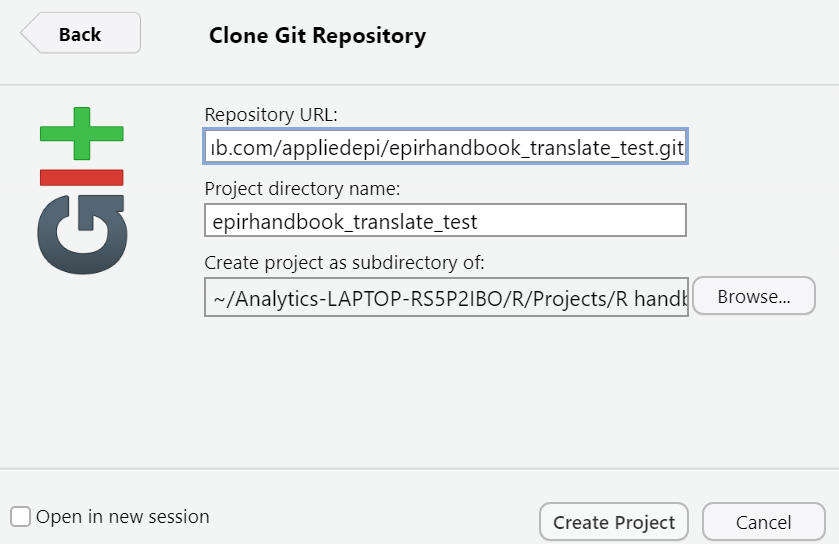
Select “Version control”



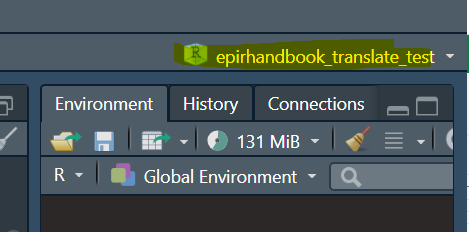
Select “Git”



Paste the “.git” address from Github that you had copied. Make sure the file directory address is set to where you want this R project saved on your computer.



Now, RStudio will download all the files and version history from this repository. This may take some time. Eventually, it will open the R project and you will see the new R Project name in the upper-right corner.

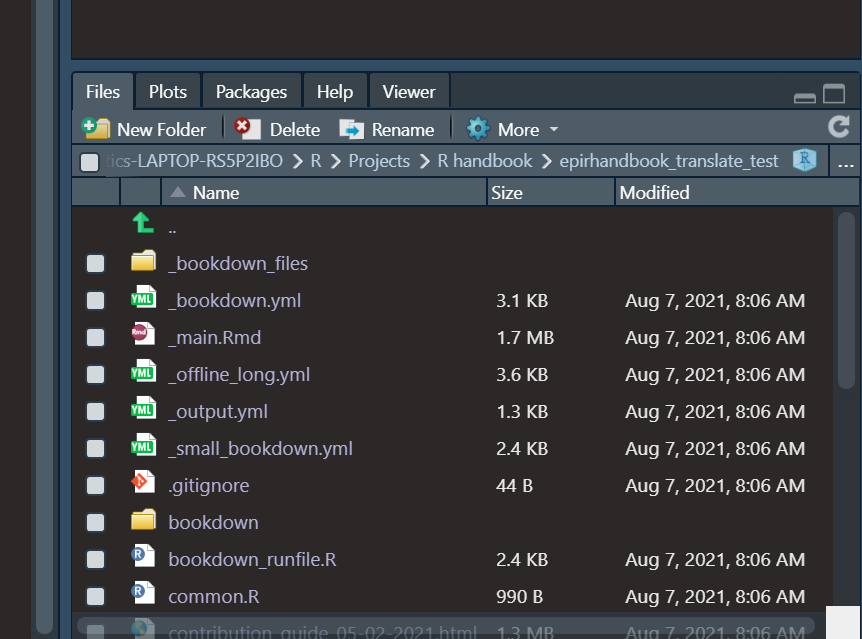


#### **Orient yourself to the repository files**

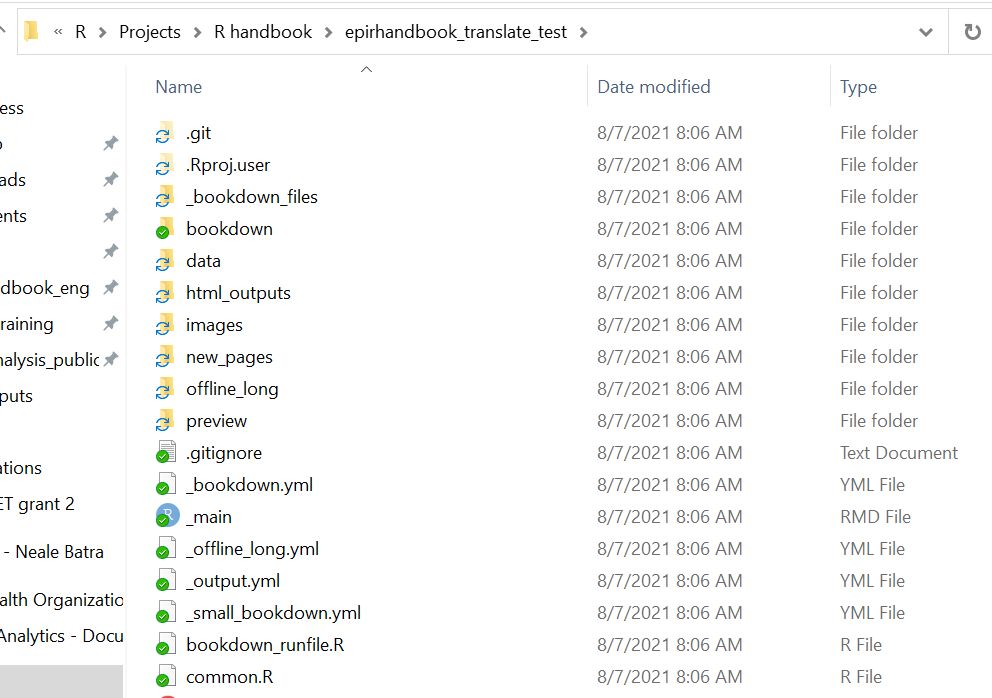
You have just “cloned” the repository. This means you have made an exact copy of all the files as they existed at the moment you downloaded them from Github.com.

Your local R project is now linked to this repository (assuming you have the permissions), such that you can make changes to the files, and those changes can be “pushed up” to the version online at Github.com.

As always in RStudio, you can navigate, open, and delete files in the R project.

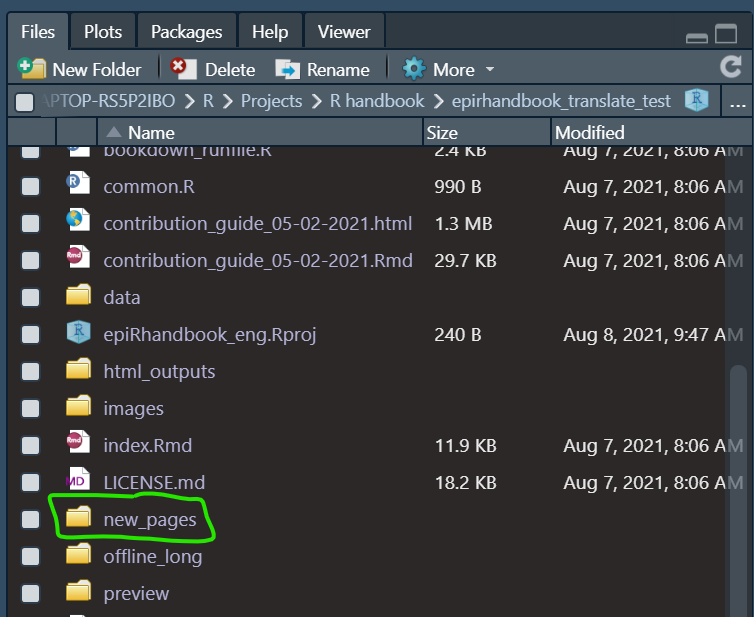


You can also view and open the files in your typical File Explorer, by navigating to where you saved the R project.

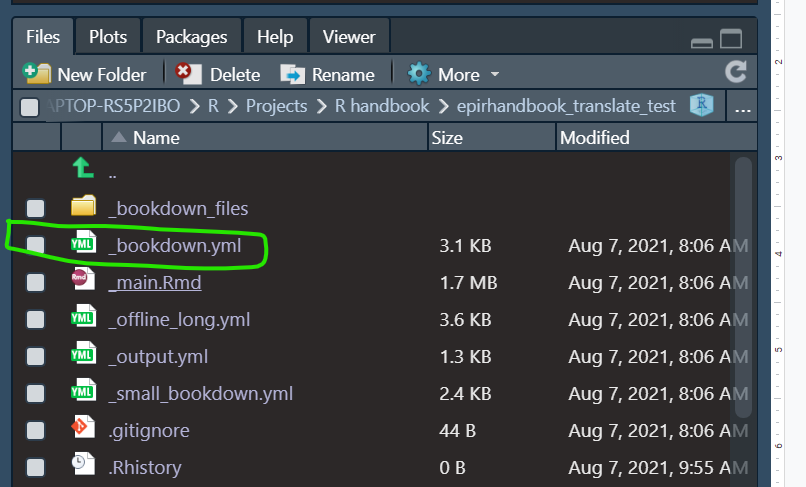


### **Finding the correct file to work on**

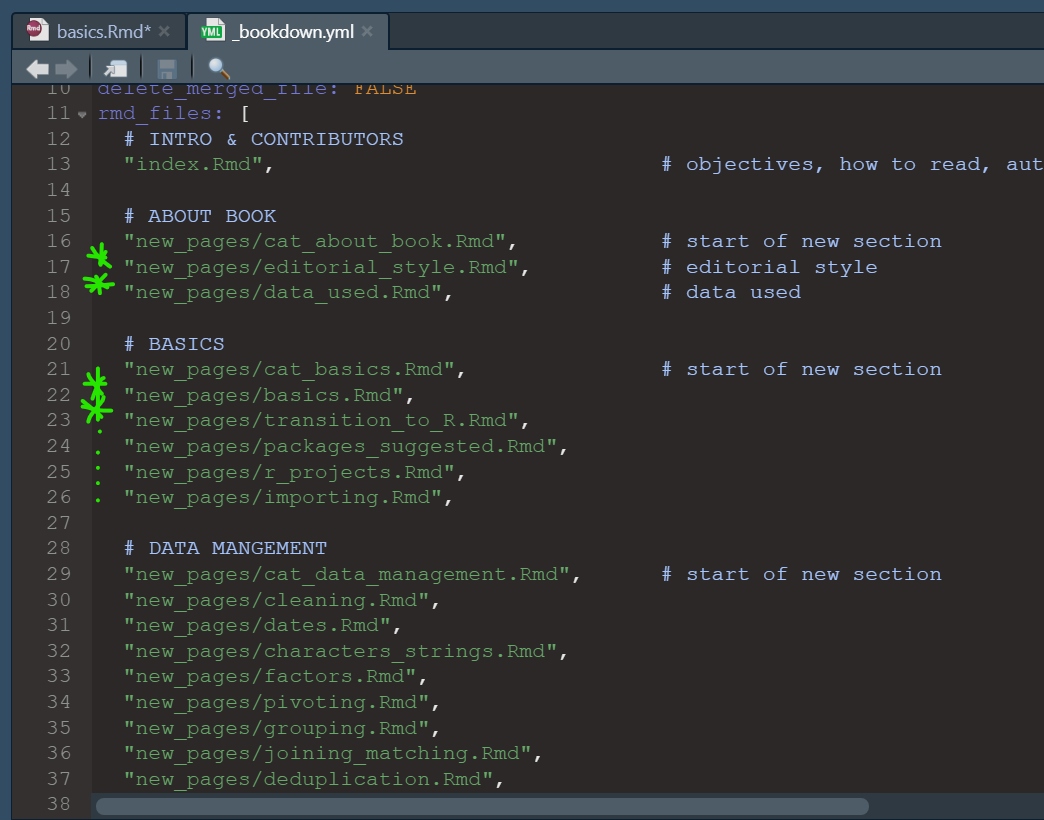
The R Markdown files are stored in the “new\_files” folder.



Inside this folder there is one R Markdown per “chapter” of the Epi R Handbook (e.g. “R Basics”).

However, the names of the files may be slightly different than the names of the chapters. If you are confused, open the “bookdown.yml” file (in the root/top of the R project).****

**This file is used to render the entire Epi R Handbook, so it contains the names of all the R Markdown files - in the correct order.**

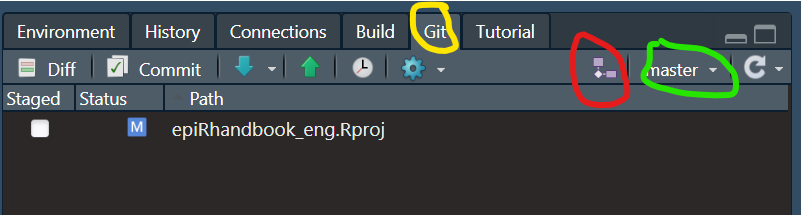
****

## **Prepare for translation**

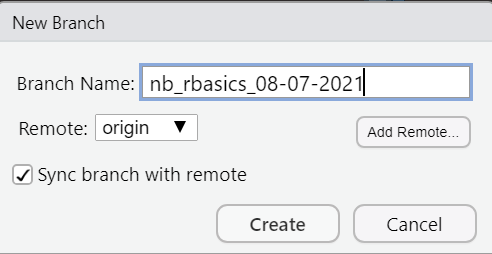
### **Make a branch to not disrupt others**

Now make a “branch” from the “master” version, in order to work on your translation. Ask your translation team coordinator for the exact procedure, but we recommend that you create one branch per “chapter”/”page” of the Handbook (e.g. for “R Basics”, or for “Transition to R”).

1. **Create a new branch for your work**. Open the “Git” pane in RStudio (circled in yellow below). Ensure that you are viewing the “Master” branch (circled in green below). THEN click the purple “Create branch” button (circled in red below).

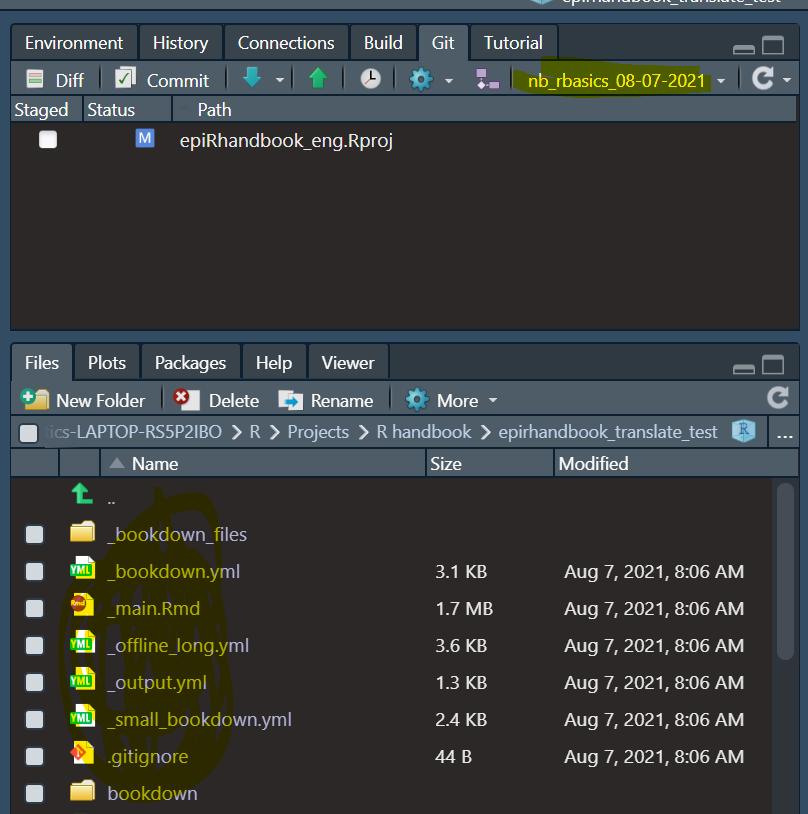


1. **Name the branch.** In the prompt window, enter a simple name for the new branch, like: “nb\_rbasics\_08-07-2021” (“nb” are my initials).

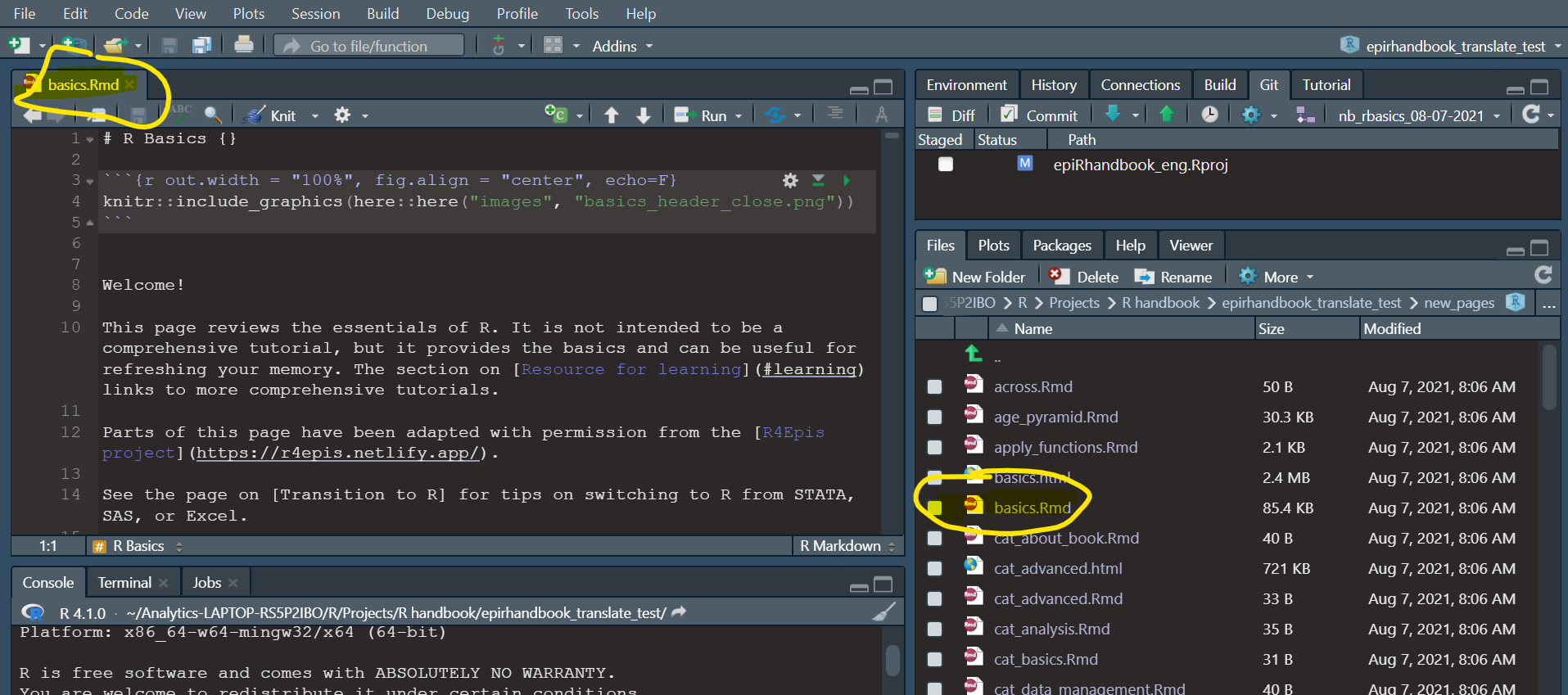


You will now see in the Git pane that the current Branch is “nb\_rbasics\_08-07-2021”. Note that ALL the files of the repository remain accessible to you in RStudio and in your File Explorer. These are now copies of all the files in the Master branch. You can edit them, but it will have no impact on the Master branch.

Regardless, we highly recommend that in this new branch, you only edit the file that contains the Epi R Handbook chapter that you mentioned in the Branch title (e.g. R Basics, in this scenario).



1. **Open the R Markdown that you want to begin to translate.** You can find it in the “new\_pages” folder of the repository. For this scenario, it is “basics.Rmd”.

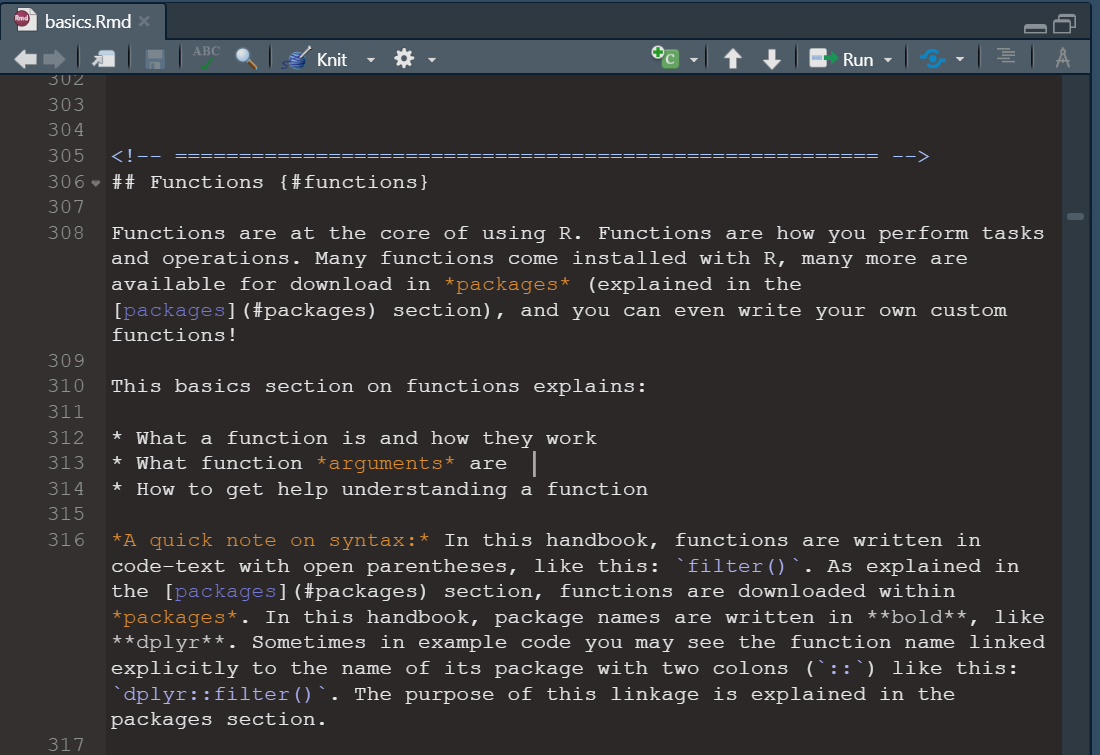


Now you can begin translating.

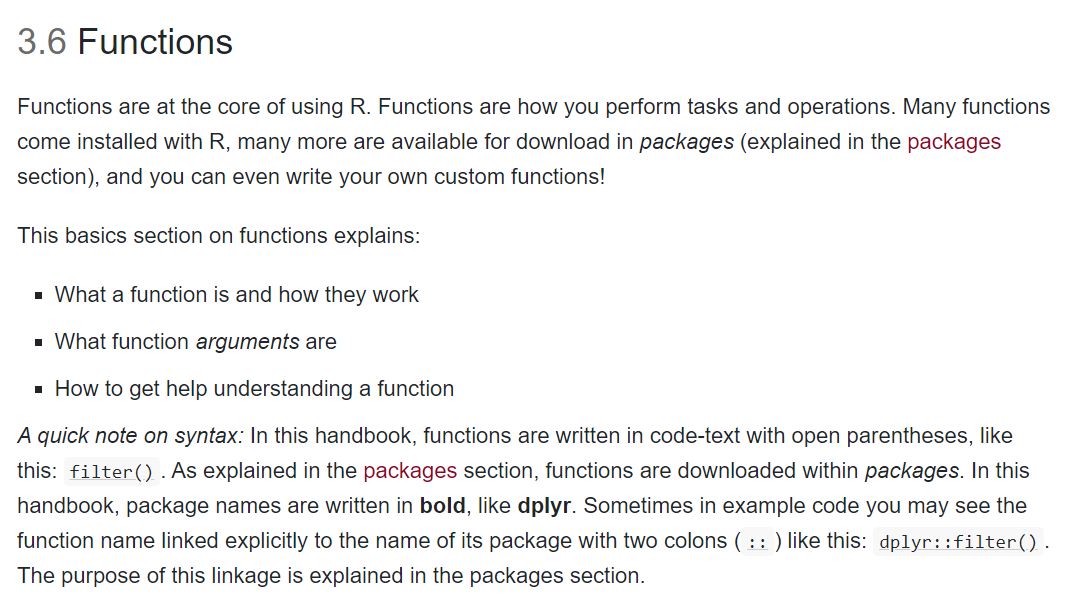
## How to translate

Let’s say that you want to translate this part of the R Markdown. It contains the following:

* A second-level header (“## Functions”) with an adjacent “tag” in curly brackets {#functions}.
* Some text explaining what an R package is
* Some parts of the text which are **bold**, or *italics*, and others that are in “code text”.
* Bullet points, created with asterisks \* and careful newlines and double-spaces at the end of each sentence/bullet.



Here is how it looks in the actual website:



### **Decide your approach for translation.**

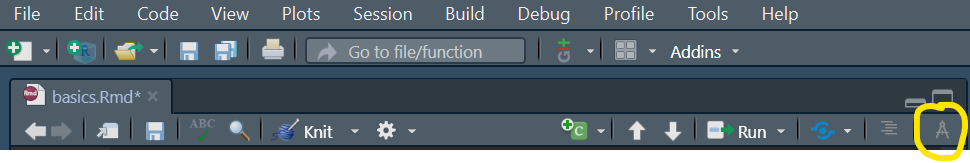
You have a few options on how to translate, discuss with your translation team lead the best approach for you:

1. Edit and translate directly into the R Markdown script.
2. Edit and translate directly into the R Markdown script, using the “Visual Mode”
3. Use a Language translator to assist either of the above (DeepL.com is very good).

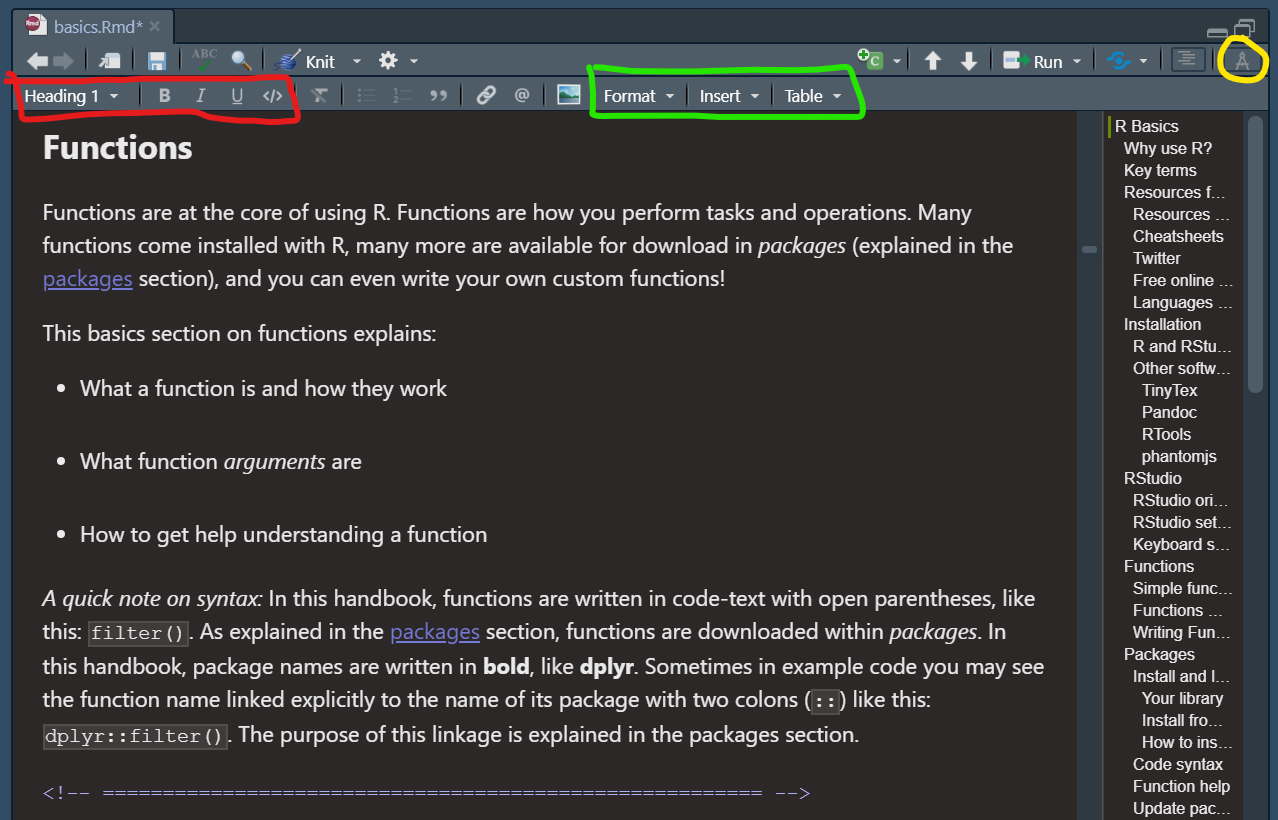
Below, we will describe option 3, with the assistance from DeepL.

### **Edit within the R Markdown “Visual Mode” with the help of DeepL.com**

When viewing the R Markdown for your chapter, click this small button in the upper-right that looks like “A” (but is actually a symbol for a “compass” for drawing”). This will convert the R Markdown script to display as *in a word processor, similar to Microsoft Word*.



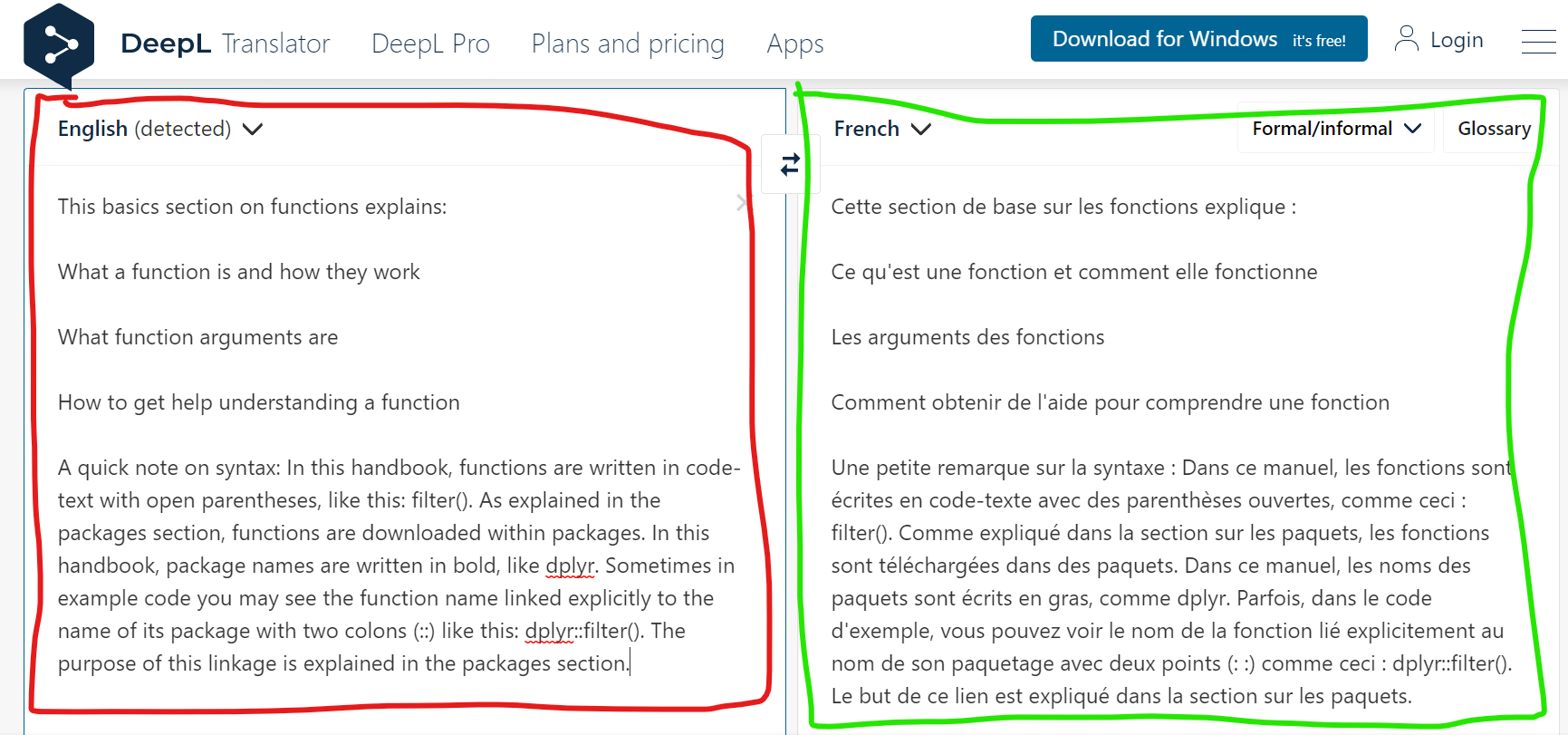
Now, this text we want to translate appears as below in RStudio. You can edit the text as in a normal Word Document. Note the tools (circled in Red) to make text a header, bold, italics, underlined, or “code-text”. You can also use the tools (circled in Green) to insert and manage Tables and pictures, although this should not be necessary when translating.



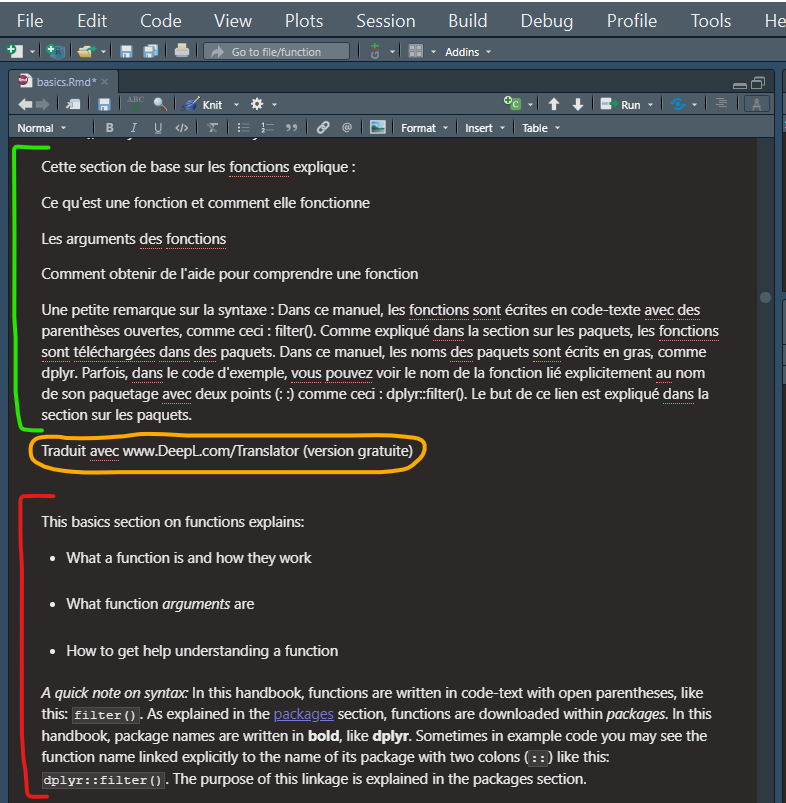
At this point, you can either translate directly into the R Markdown as shown, retaining the links and other formatting. Or, you can use the assistance of an automated translator like Google Translate or DeepL.com.

If using an automated translator, your procedure may work like this:

1. Copy a paragraph of text from the R Markdown and paste into DeepL.com.

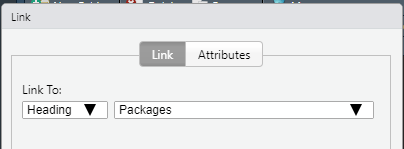


1. Edit the translation so that it is correct by your standards.
2. Copy and paste the translated text back into the R Markdown, so that you can see both the formatted English and the translated.

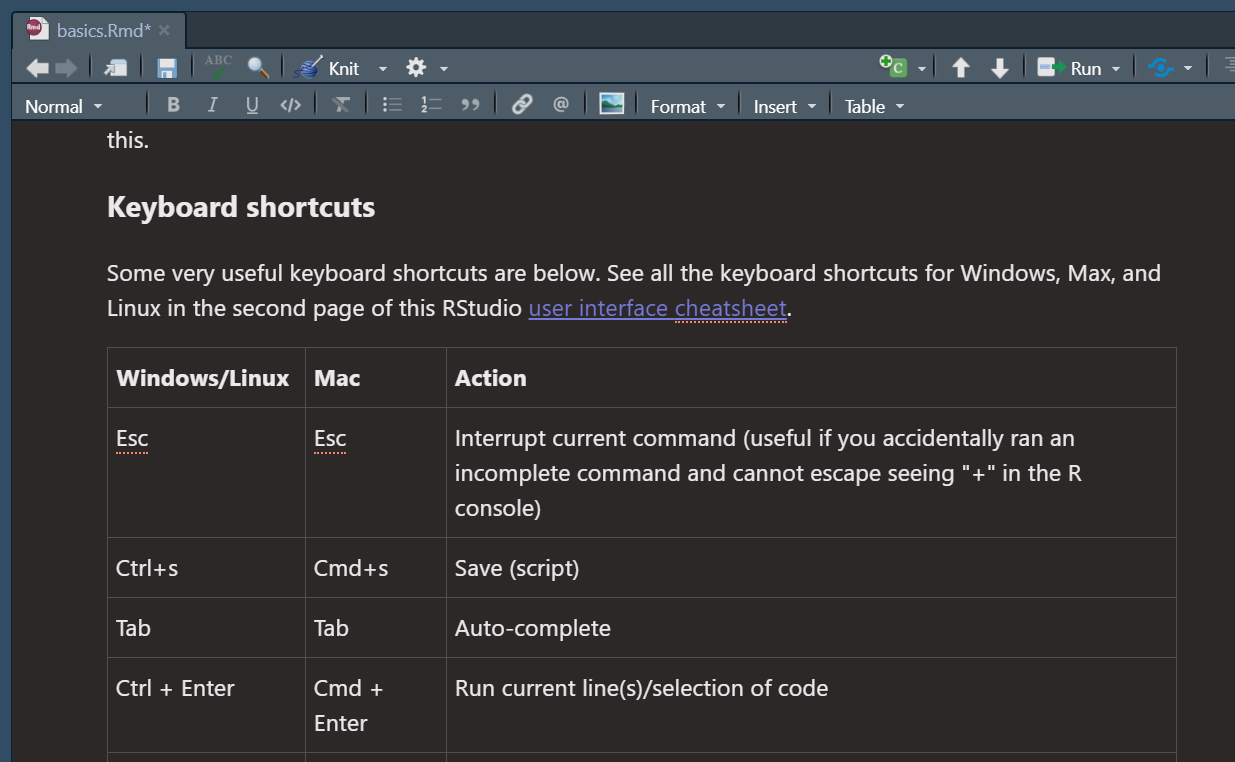


1. Be sure to remove any “stamps” from a translating service (e.g. “Translated by Deepl.com” - see this circled in orange above.)
2. Now, use the Visual Mode tools to align the formatting of the translated text to that of the English. For example, to re-establish bullet-points, or make certain words italicized, bold, or “code-text”.

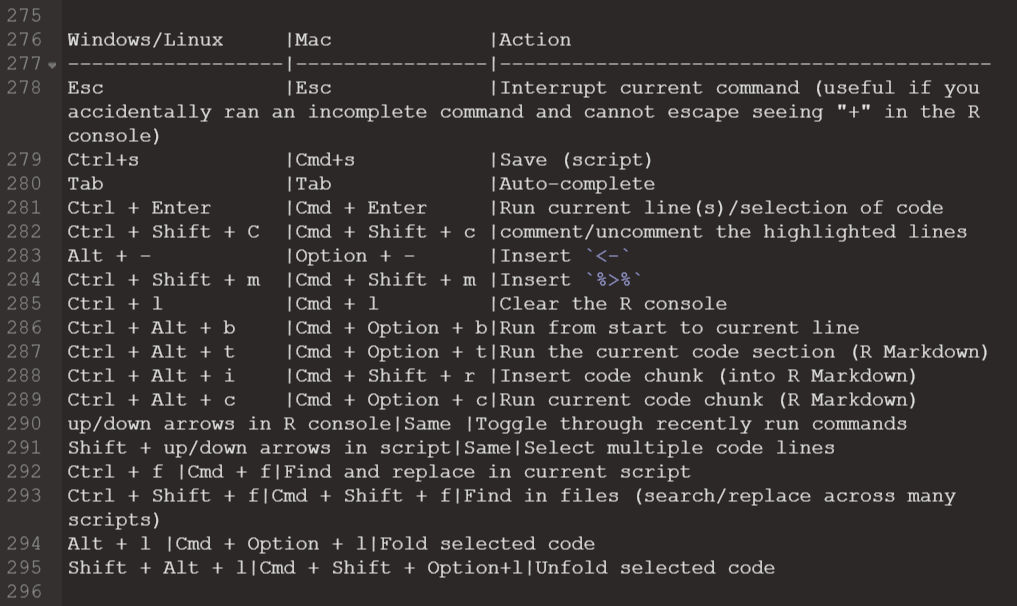
* **Bold** words are usually the name of R packages. These must be in **bold** for consistency. If using Visual mode, simply highlight the text and click the bold button. If you are not in Visual Mode and are working in the raw R Markdown script, ensure there are two asterisks on each side of the word or phrase. For example: \*\*dplyr\*\*, or \*\*this is a very important point\*\*.
* *Italics* are easy in Visual mode, but if you are in the raw R Markdown they are are created with one asterisk on each side. These are usually emphasized words, like “It is \*very\* important that you run this command.”
* “Code-text”. Any R function or in-line code (code in the text - not in an R code chunk) should be in “code-text”, like below.  Note that the names of R functions should have two empty parentheses afterward, like select() or filter() or plot(). This informs the reader this is a function. You can create this “code-text” in Visual mode by clicking “</>” button, or in the raw R Markdown by putting “back-ticks” around the text. Like this “the function `sqrt()` calculates the square root of a number”. Note that backticks are DIFFERENT than single quotation marks.
* If there are links that are lost during this copy-paste process, you can restore them using the “link” button or “Insert -> Link” drop-down. A window will open like below, and you can select the appropriate Heading to link to. Links are delicate, so ask your translation team lead if you encounter difficulties. Also see below more trouble-shooting tips on linking to other pages.



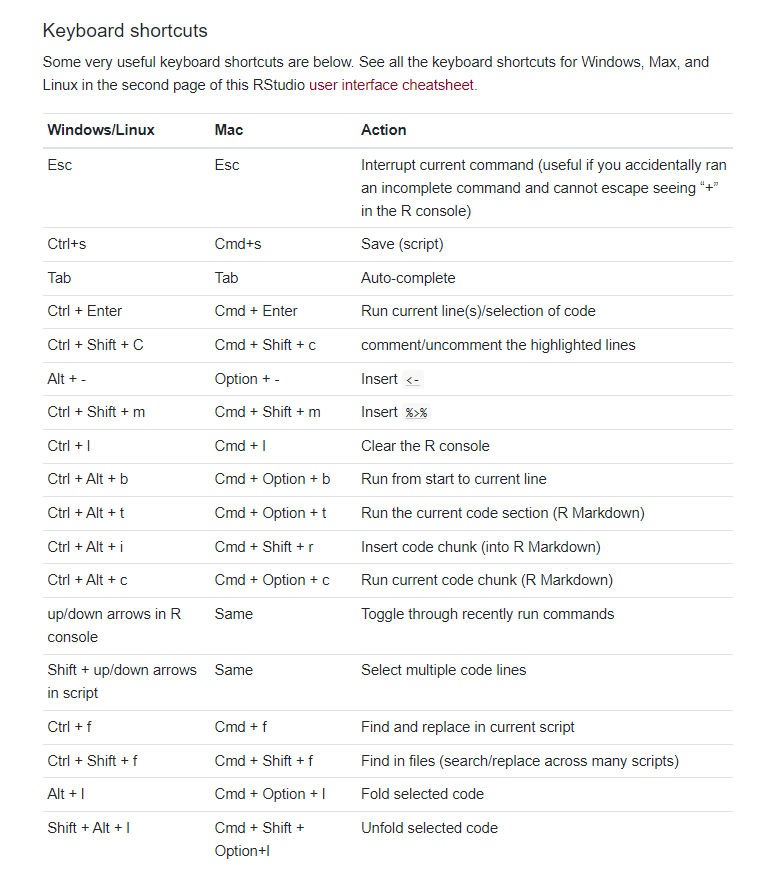
* Tables in the text can easily be edited in Visual Mode.



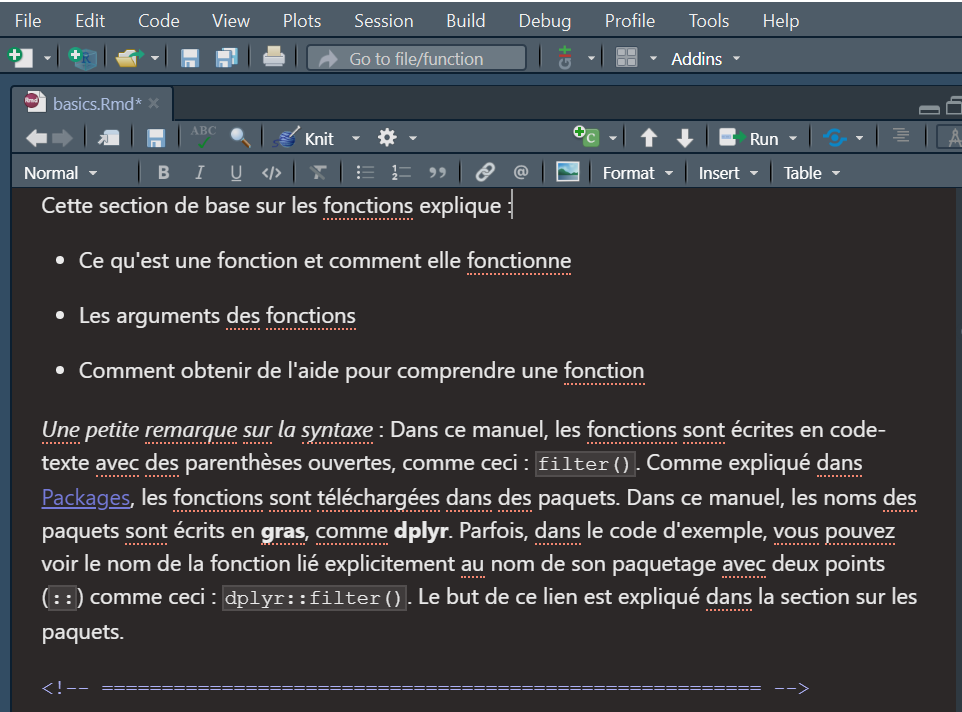
If you are editing a table in the raw R Markdown, adjust the text carefully and do not adjust the newlines. You can add longer text such that the “bars” | do not line up - this is OK if they do not line up. What is important is that the newlines remain intact (a newline at the end will produce a new table row. In R Markdown the code may look like this:



And in the website the top of the table looks like this:

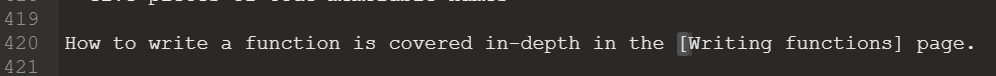


1. Voila! The text is translated and has the correct formatting! Delete the english, *save your work*, and move on to the next paragraphs.

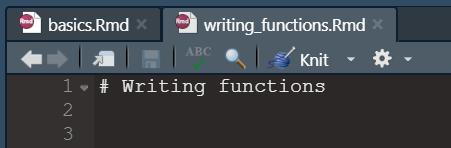


1. *A note on Chapter titles and internal links*

Links between pages are done in the raw R Markdown via this syntax below. Note that this methods requires that the text in the brackets be EXACTLY the title of the page being linked to.



The title of the Writing Functions page looks like this, in the writing\_functions.Rmd. The title is created by one Hash (#) at the top of the Rmd

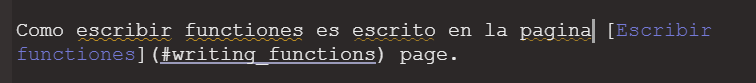


Sometimes, we have found that these links do not work in the translations. In this case, please use this alternative method to linking between pages using the “tags”....:

1. Rename the page being linked to, but add a “tag” in english afterward, in *curly* brackets with no spaces



1. In the page that has the link, write this to make the link according to the english tag.

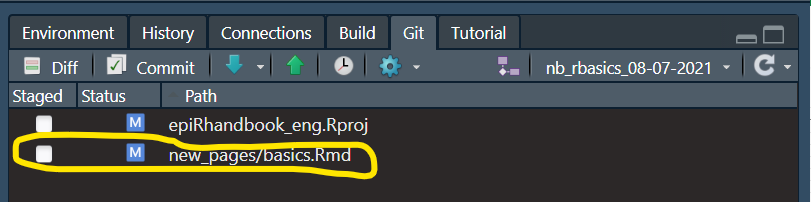


This way, the text and the title are translated, but the link is made via the English “tag”.

### **Commit and Push your changes**

Every change you make is tracked locally by Git. You can bundle these changes together into a “commit” with a short message describing the work you did. Each commit can be reversed if needed. It is best to make many small commits rather than a few large commits.

When you save your R markdown file with changes from the original, it will appear in the Git pane. This means the file has been changed.

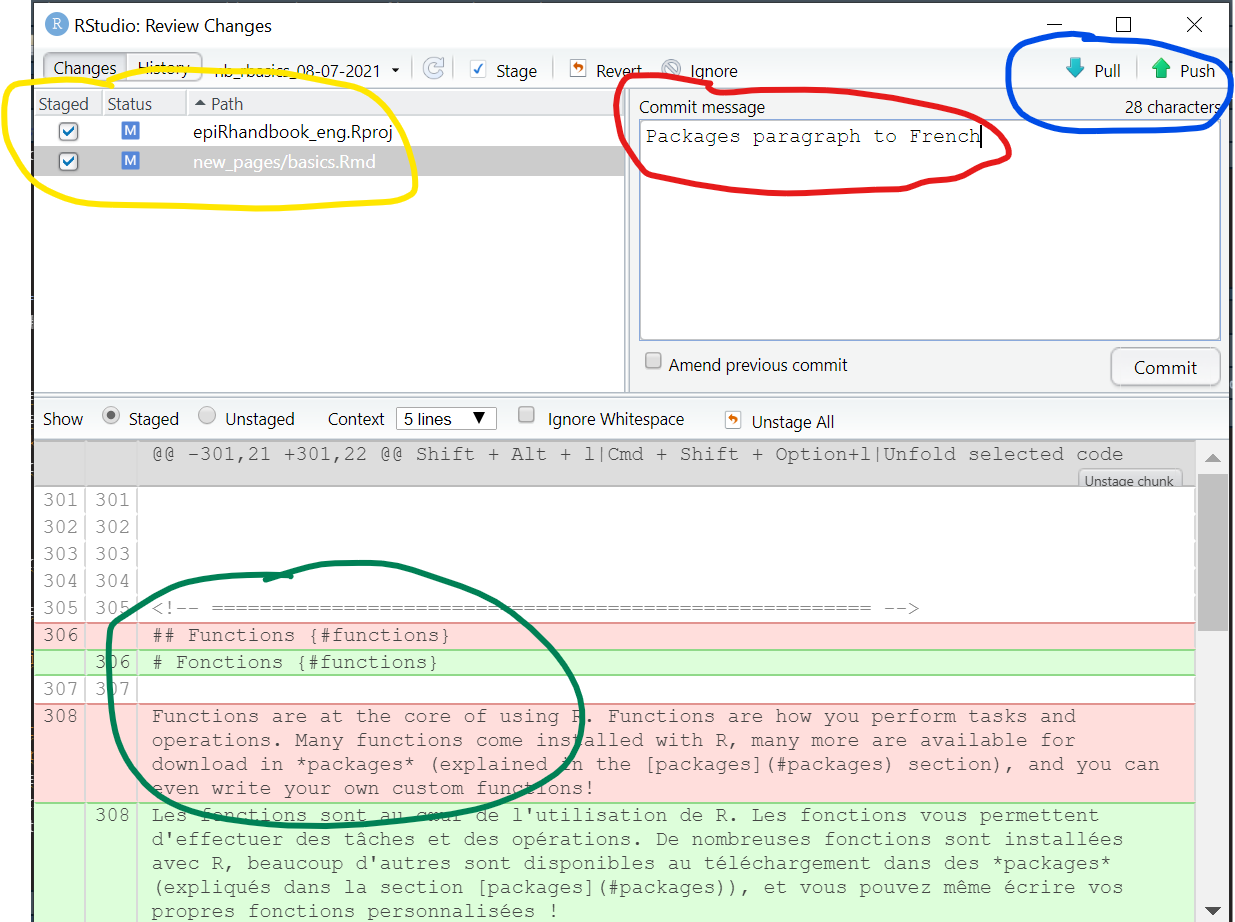


Now you have two paths forward: You can work in RStudio completely, or you can use Github Desktop to assist you.

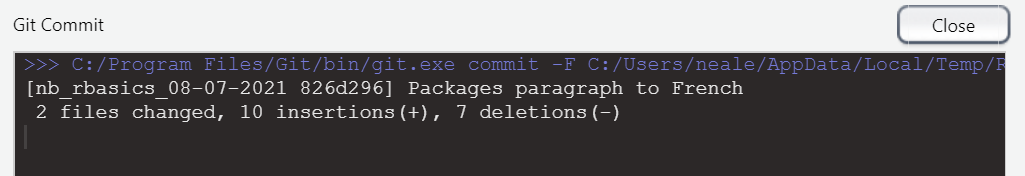
For today, we will show how to work entirely in RStudio. Using Github Desktop offers faster speeds. Ask Neale or Alex or your translation team lead to help if you want to learn Github Desktop.

Commit your changes by clicking the “Commit” button in the RStudio Git pane. A new window will open.

* In the upper-left (circled yellow below), make sure all the boxes are checke (“Staged”) for the files that you want to include in the commit (generally, include all the files that are shown).
* Write a brief commit message describing what you did (circled in Red below)
* Review the changes you made (circled in green below)
* Then click “Commit” button, below the commit message.



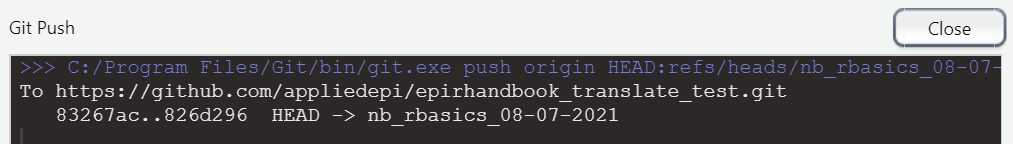
You will see a window pop up to confirm the commit. You can close this.



Now you can PULL and the PUSH the changes up to Github’s version of the “nb\_rbasics\_08-07-2021” branch. *Until now, the changes have only been existing on your local computer’s version of the “nb\_rbasics\_08-07-2021” branch.*

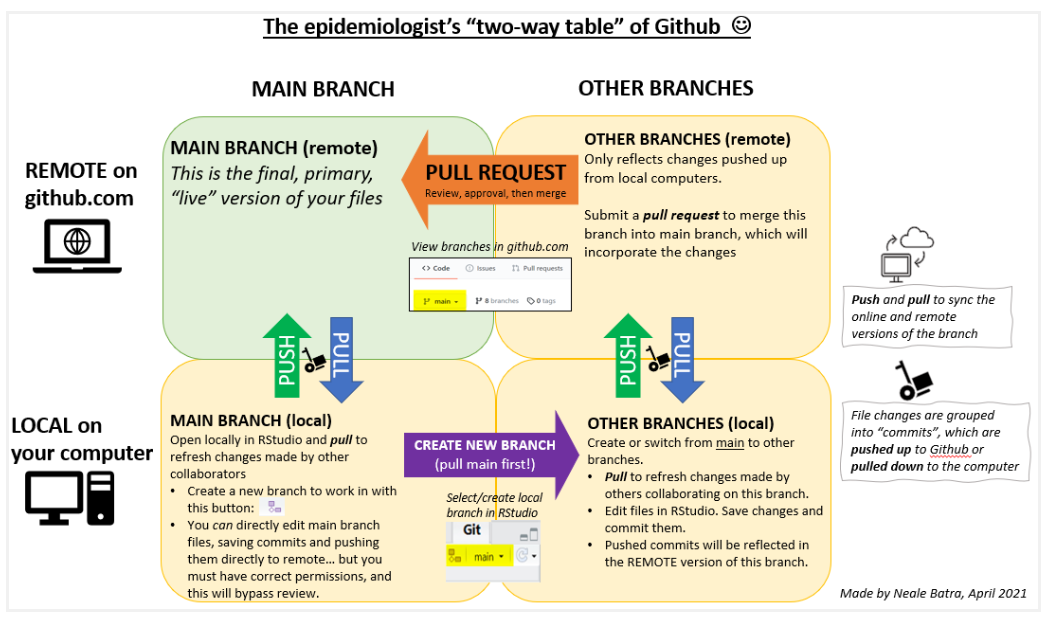
1. Click the “PULL” button (the DOWNWARD arrow). ALWAYS DO THIS FIRST.
2. Click the “PUSH” button (the UPWARD arrow).
   1. This will push your commits up to the translated repository on Github.com, to the branch that you have been working on.

You should get a confirmation message like this:

**

*If you get errors about authentication, ask your Team Lead for help. You may need to login to Github and confirm your account information.*

*Here is a diagram to orient you if you are confused:*

**

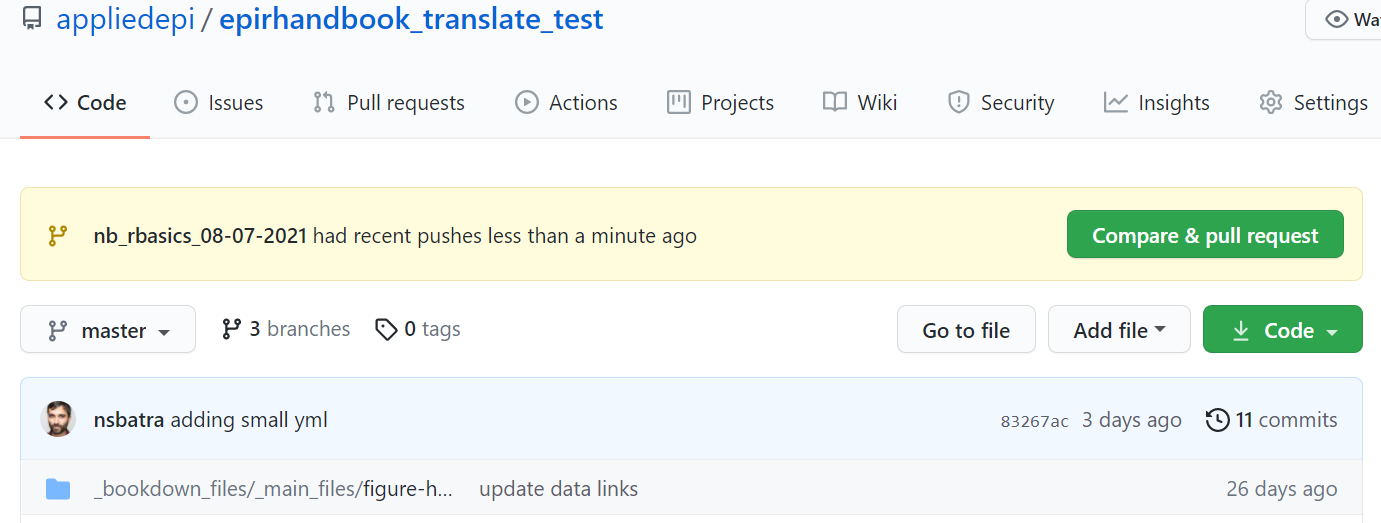
*If you are still confused try* [*reading the Epi R Handbook chapter on using Github and R.*](https://epirhandbook.com/version-control-and-collaboration-with-git-and-github.html)

### **Submit a Pull Request**

Once you have made many commits and pushed up to Github - enough that have finished your page and you want your team lead to review your work, you will create a “Pull request” in Github to move the changes from your branch into the “Master” version of the files.

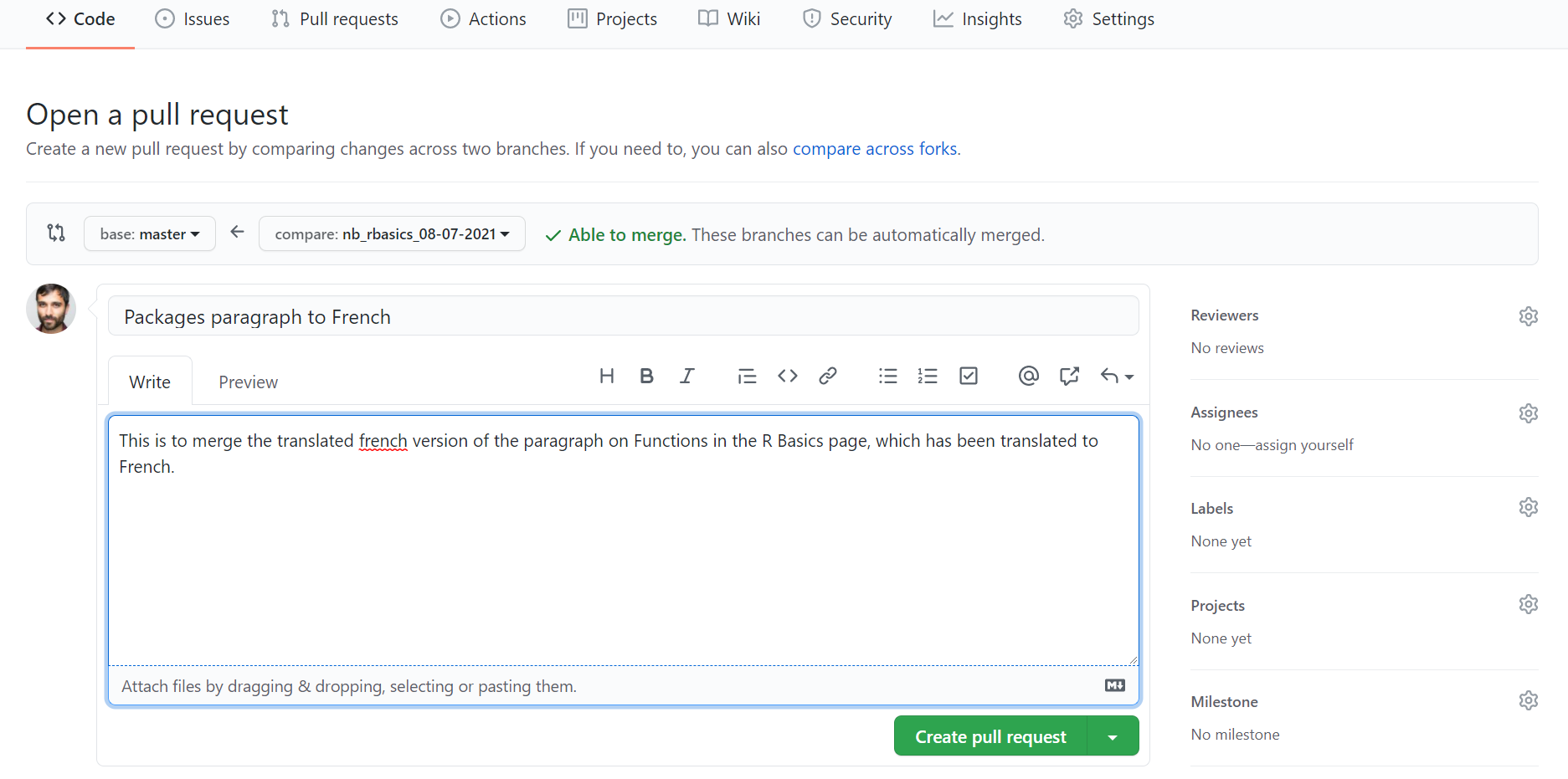
Go to Github.com and open the translated repository.

If you go there fast, you will see this banner noting that your local changes were received online, and suggesting that you make a “Pull Request”. Click the button Compare and Pull Request..



The window will now allow you to type a detailed message explaining why you want to merge your changes into the “Master” version files of the translated repository. Remember, until now, your changes are only reflected in your “branch” of the master version.

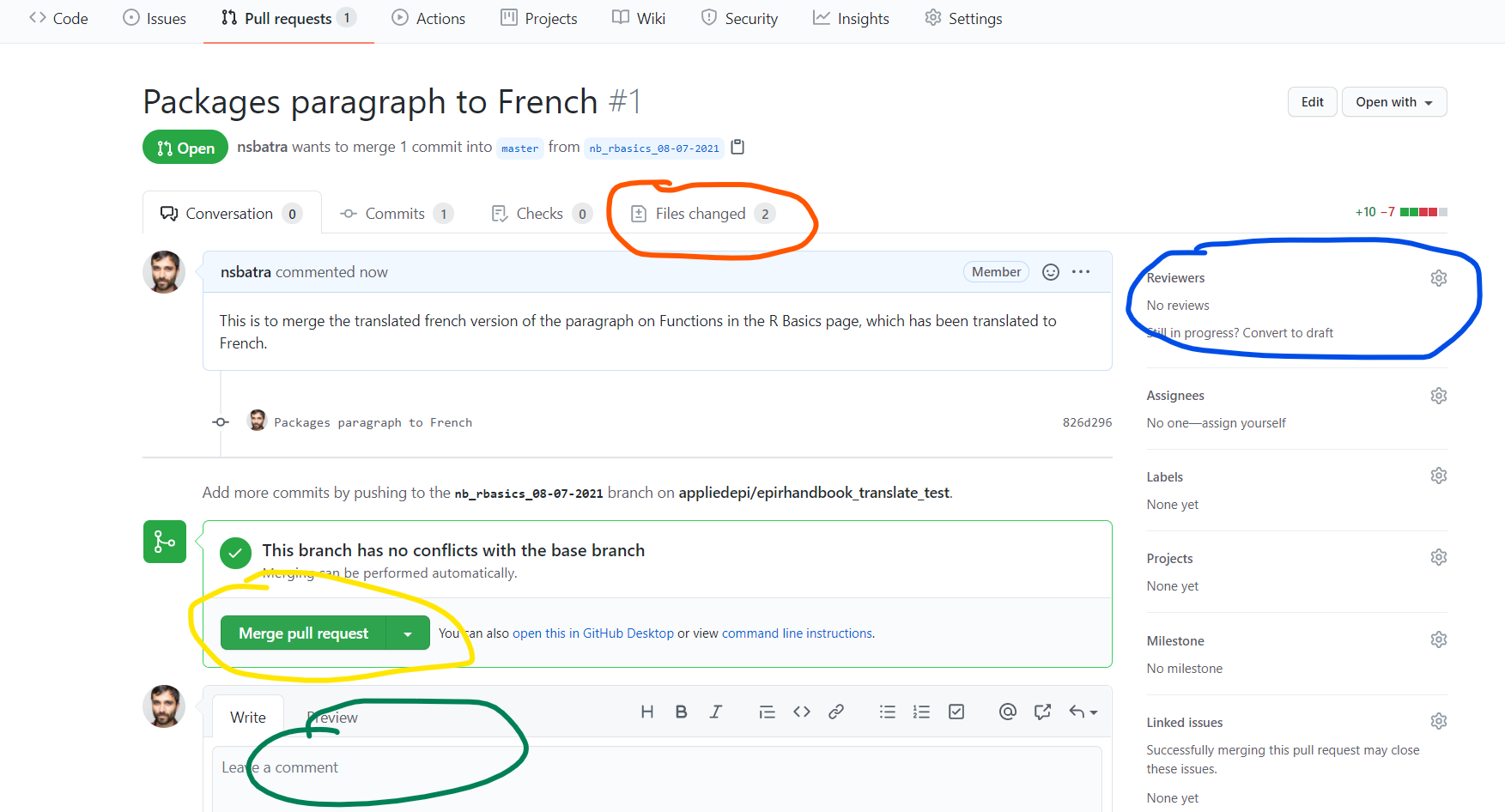
When satisfied, click “Create Pull Request”. Note that usually, you could not create a pull request for such a small amount of translation.



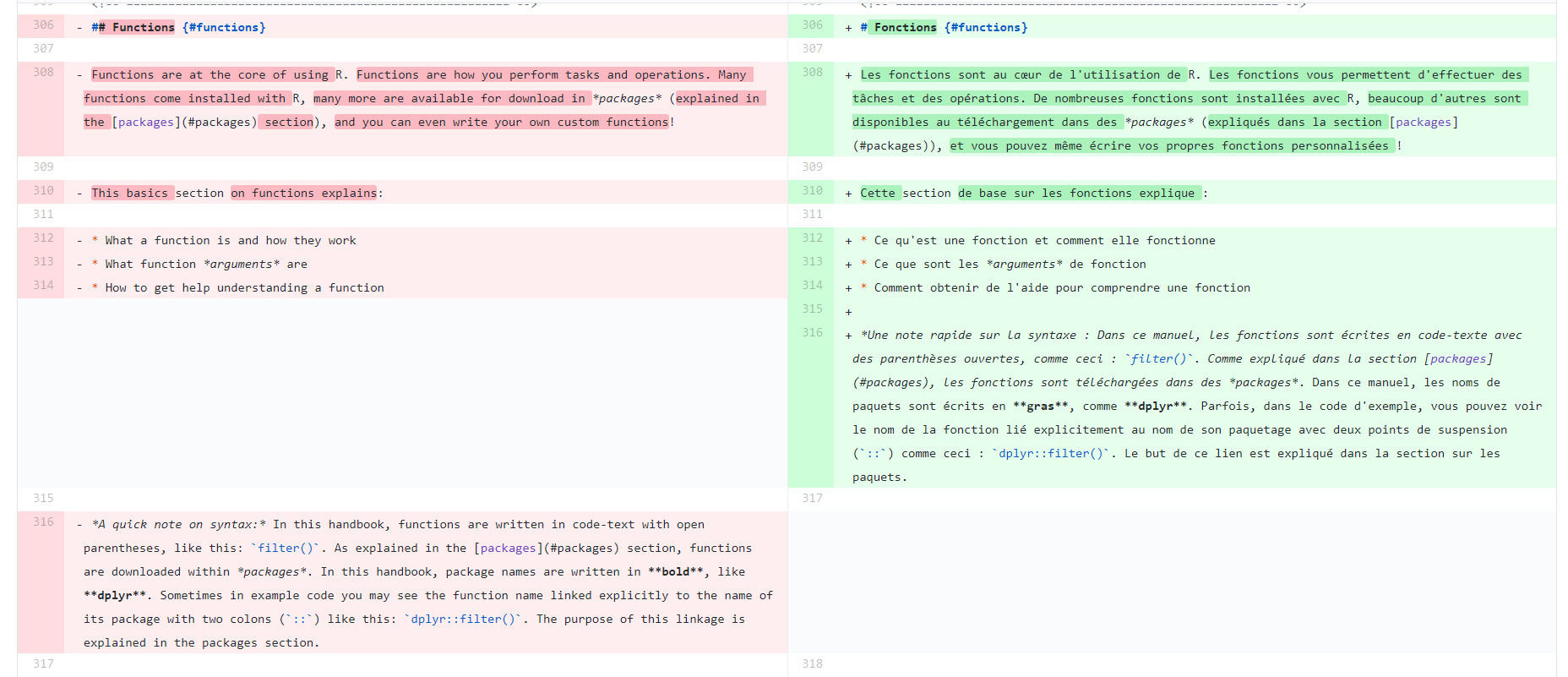
You will now see this summary of the Pull Request.

Add your translation team lead or others to the “Reviewers” (circled in Blue below). Be sure to have someone review your work before Merging the Pull Request. In fact, you may not have permission to merge until it has been reviewed.

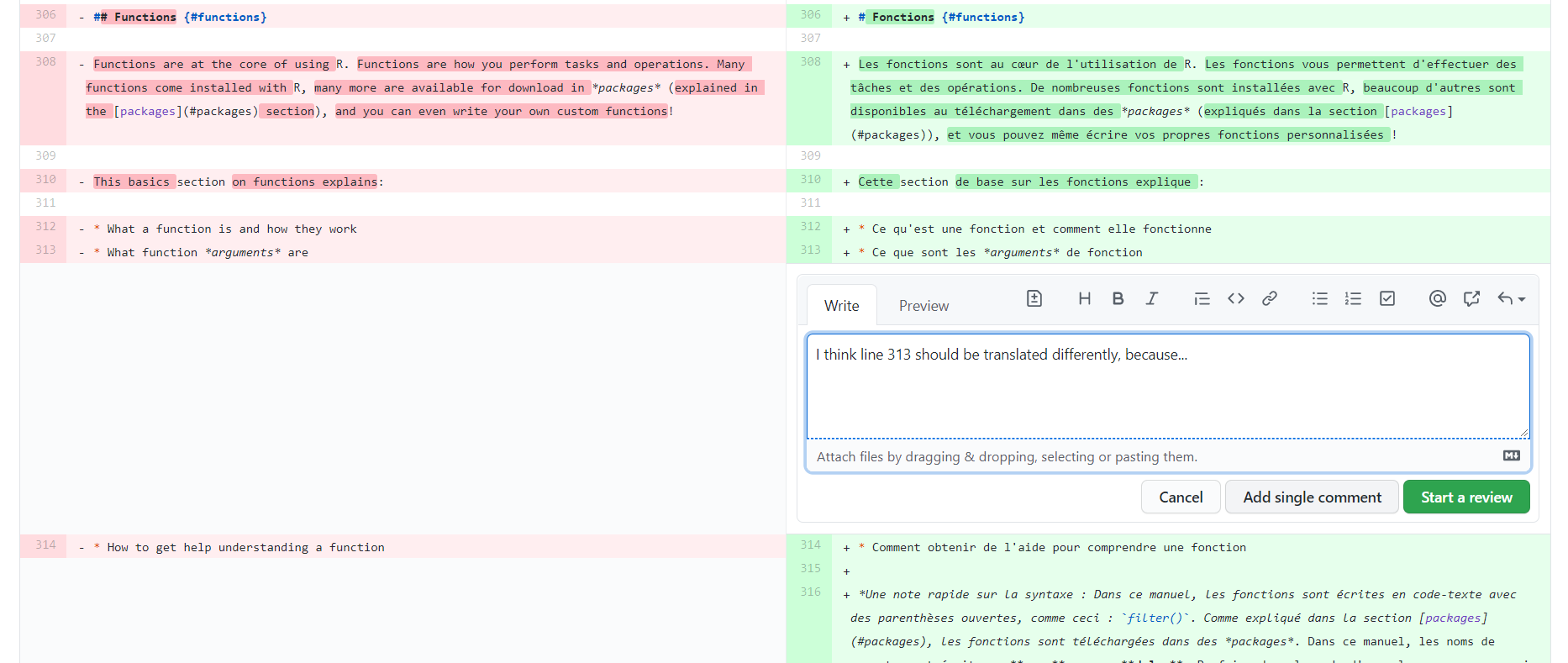
You can also write notes and tag people with “@” symbols in the comment box (circled in green below).

**

You and others can review the “Files changed” tab (circled in Red above). Here you can review the translations side-by-side (Red is English, Green is the French). This is very helpful.

**

*If you hover over the Green text, you will see a “+”. If you click this, you can insert a comment specific to a certain line. This feedback will appear in the main comments section of the Pull Request, and others can respond with feedback.*

**

*When your changes have been reviewed and accepted, ask you Team Lead what to do next. They will control the Merging into the Master branch.*

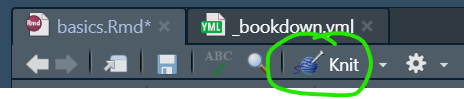
*You can make more branches, or keep working on the same one for the*

*Note that you can PUSH your changes up without initiating a pull request*

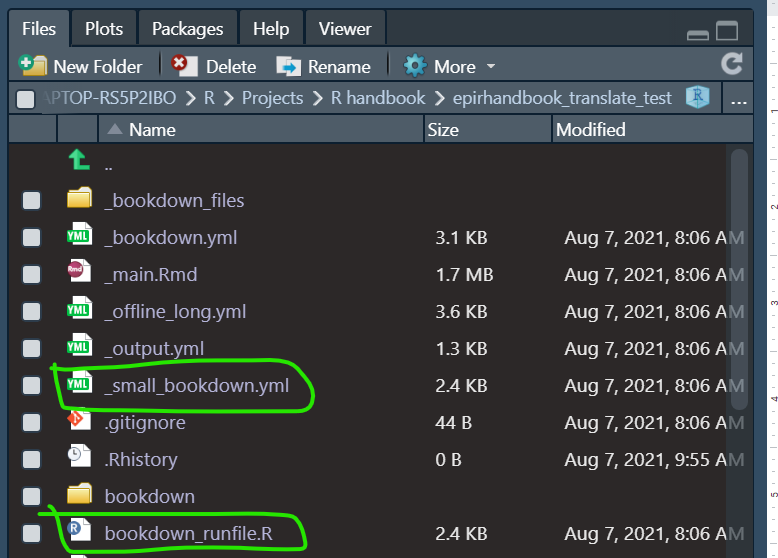
## How to “Preview” the page

You have three options:

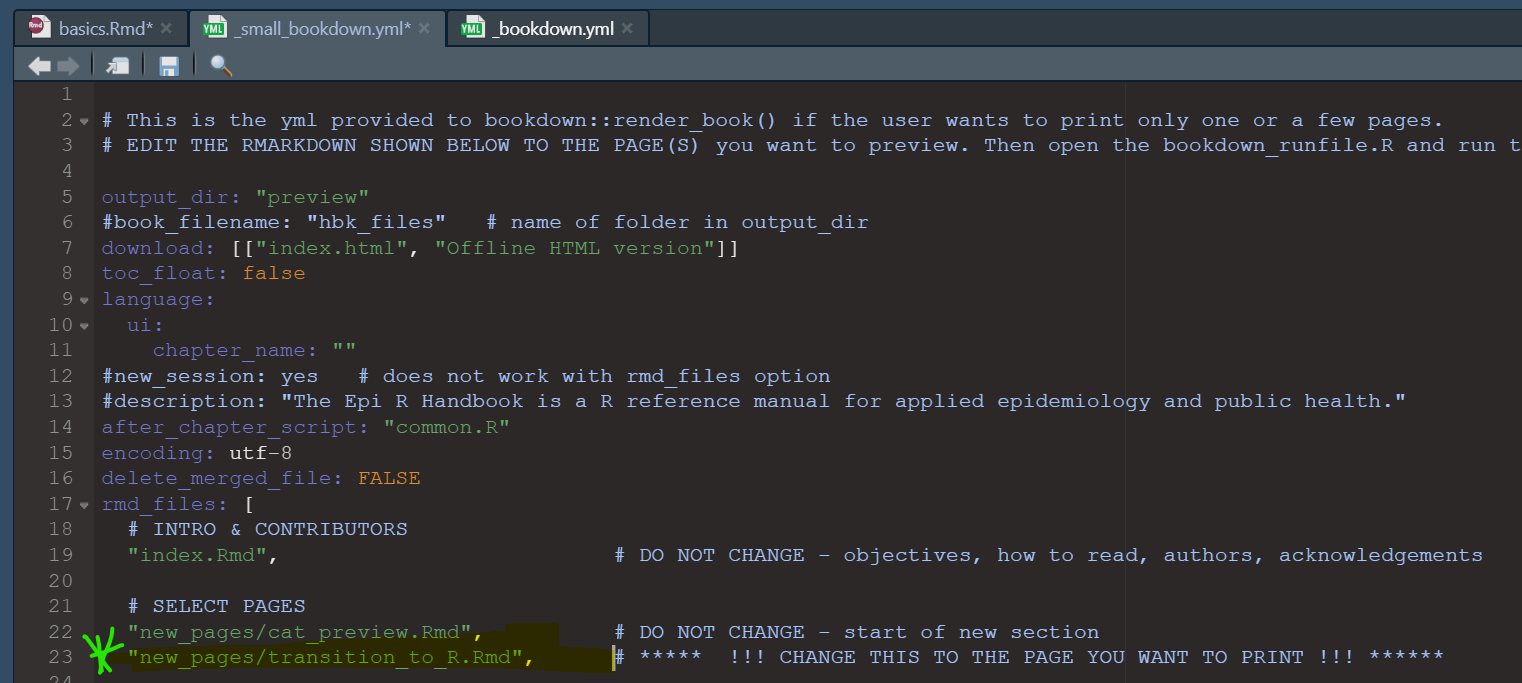
1. View the R Markdown with “Visual Mode”, as described above. This is a very simple view of the end product.
2. Press the “knit” button. This will render just this R Markdown page as an HTML. But it will not have the exact formatting as the website.



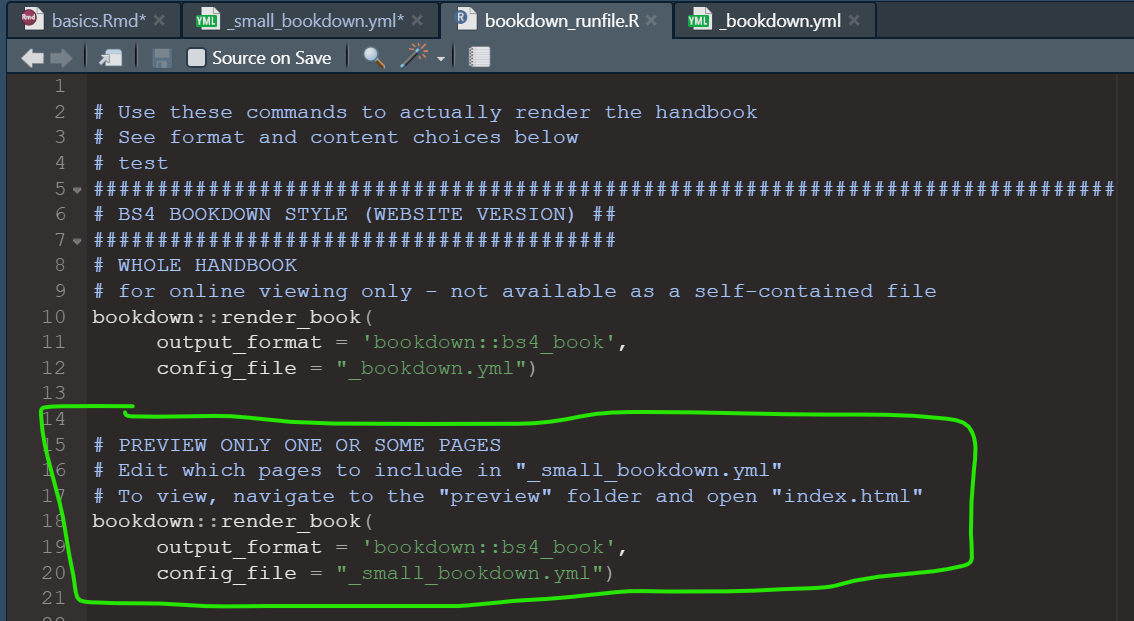
1. Render the page as it will display on the website, using the “\_small\_bookdown.yml” file and the “bookdown\_runfile.R” in the root/top folder.



First, enter the “\_small\_bookdown.yml” file and edit the R Markdowns listed. Just have one or a few R Markdown files.



Now, go to “bookdown\_runfile.R” and run the second command. This will render just the pages that you listed in the “\_small\_bookdown.yml”.



Note that sometimes when you run this command, R will try to run the ENTIRE handbook. If this happens, just stop the running with the STOP sign symbol, and run the command again. This has worked.