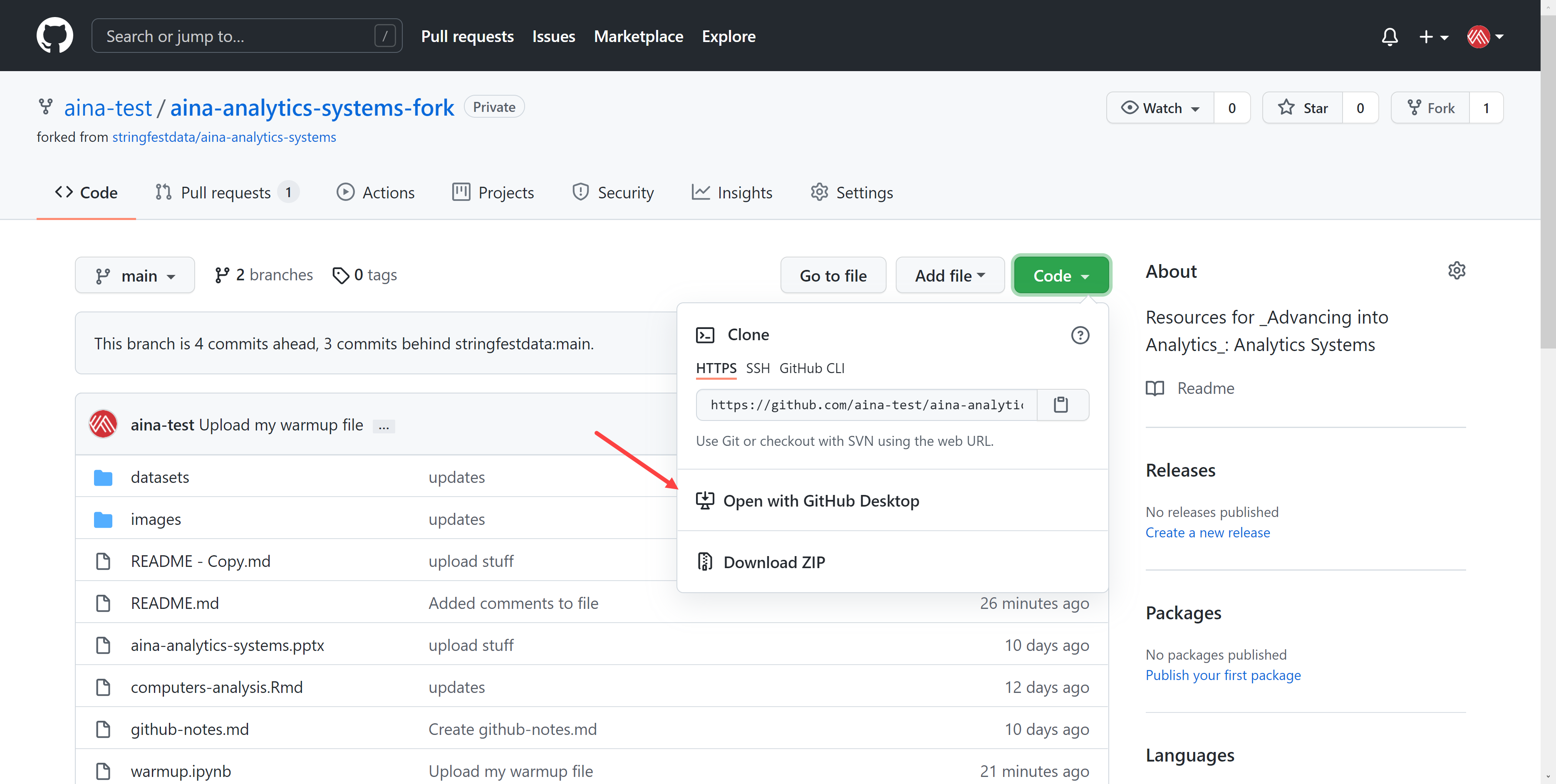
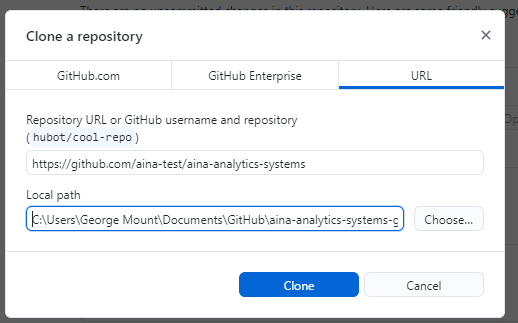
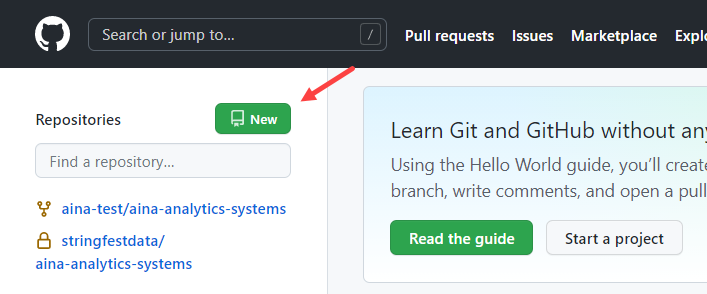
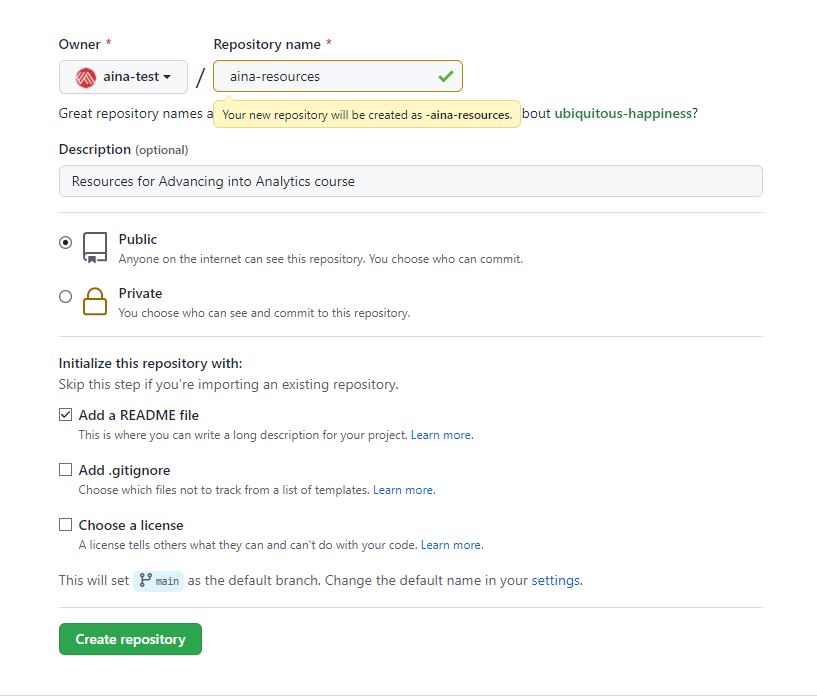
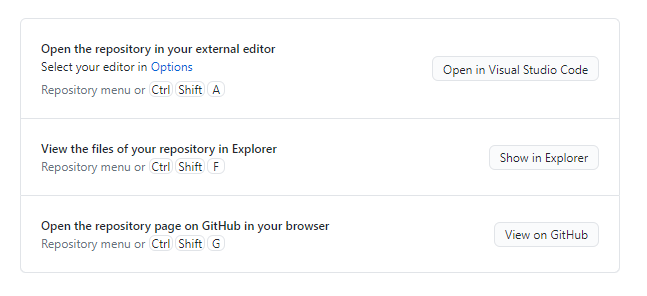
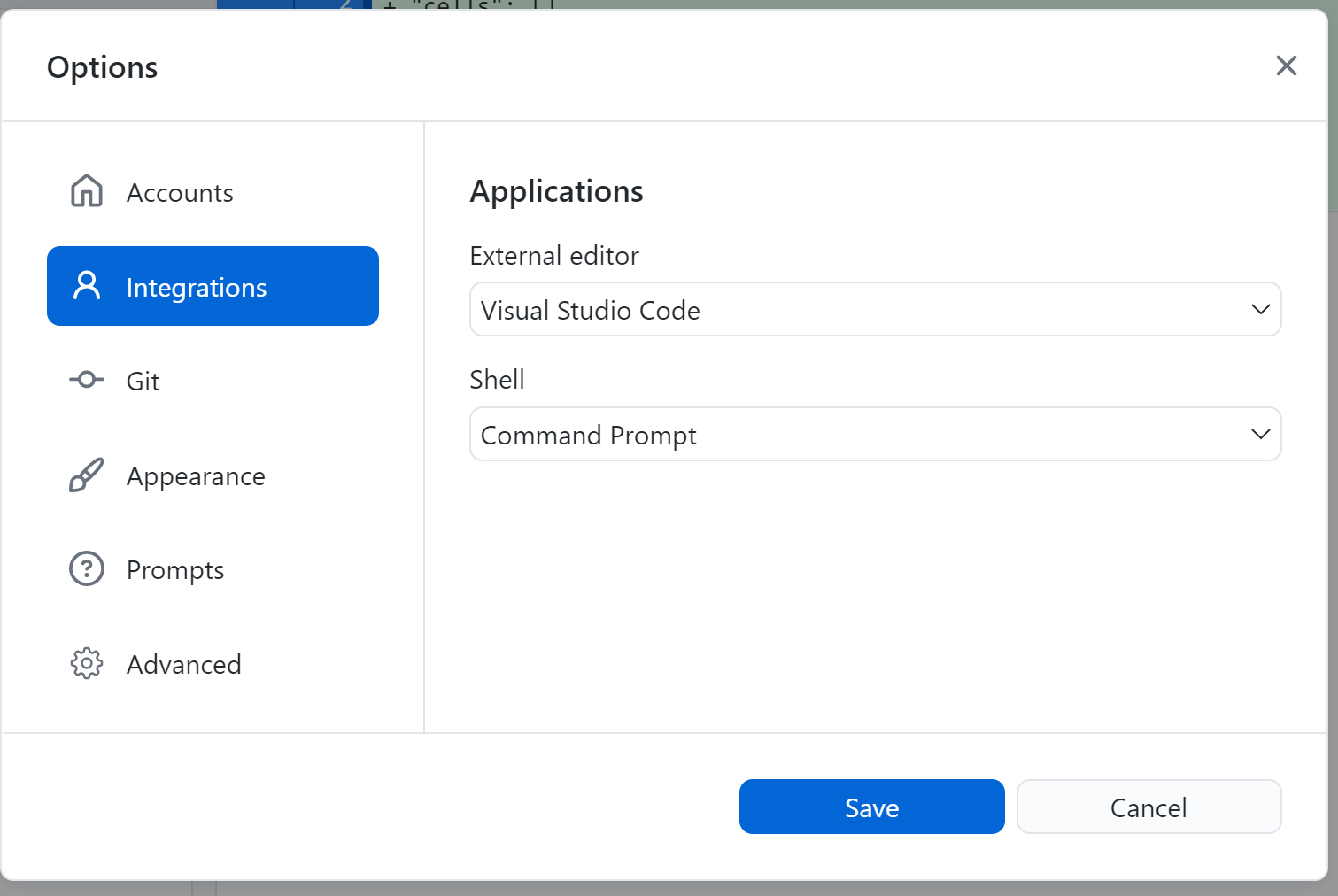
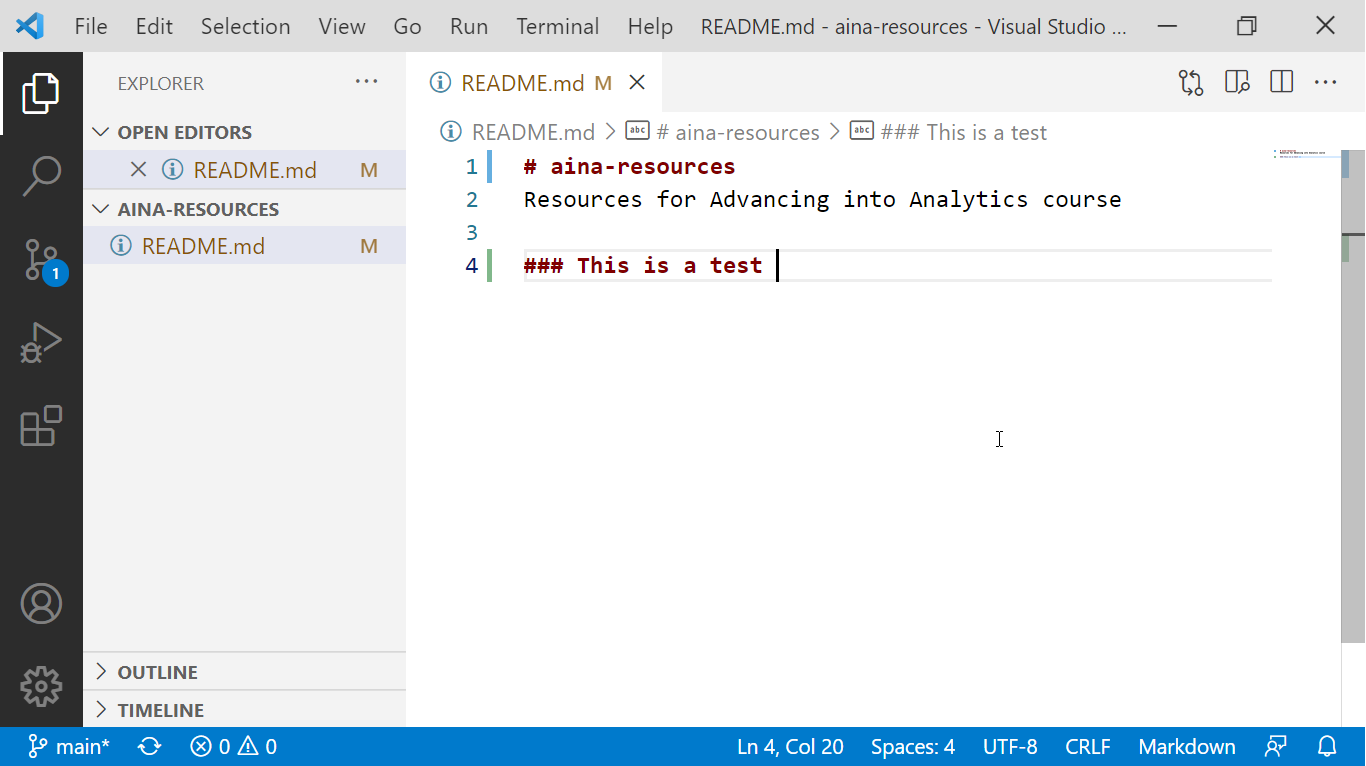
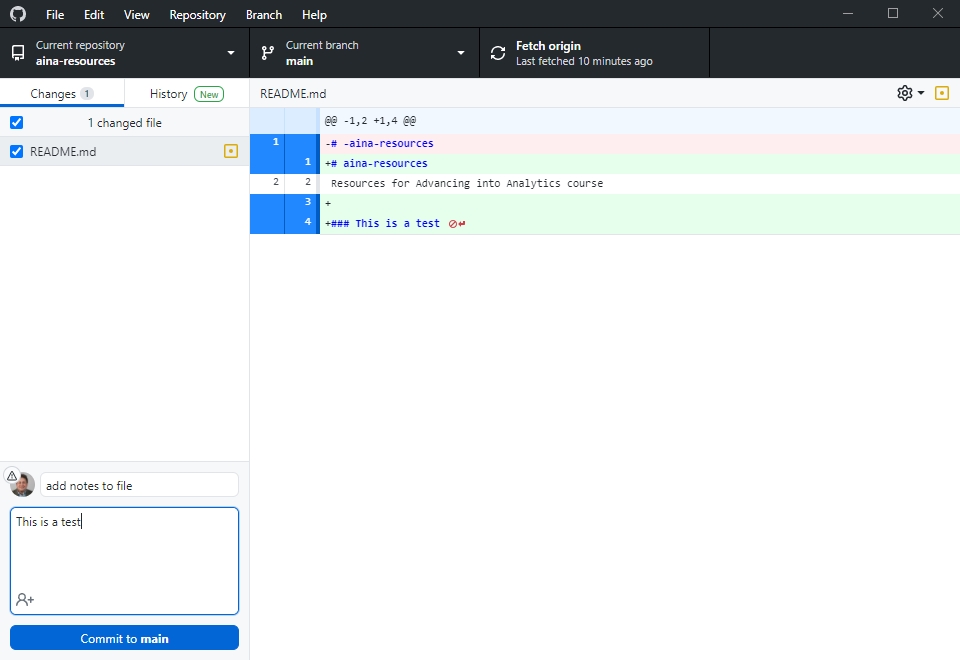
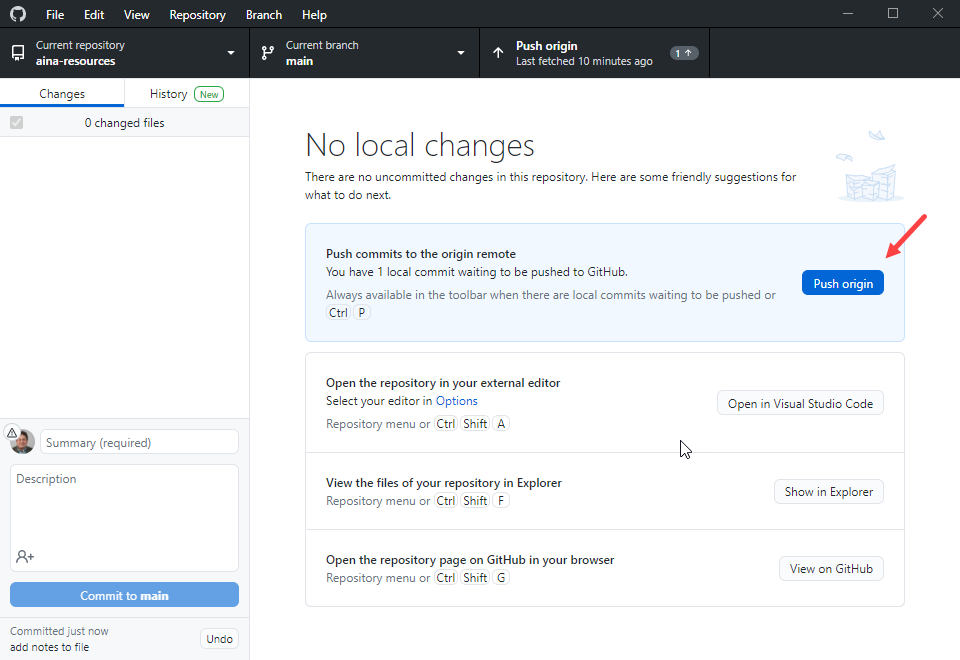
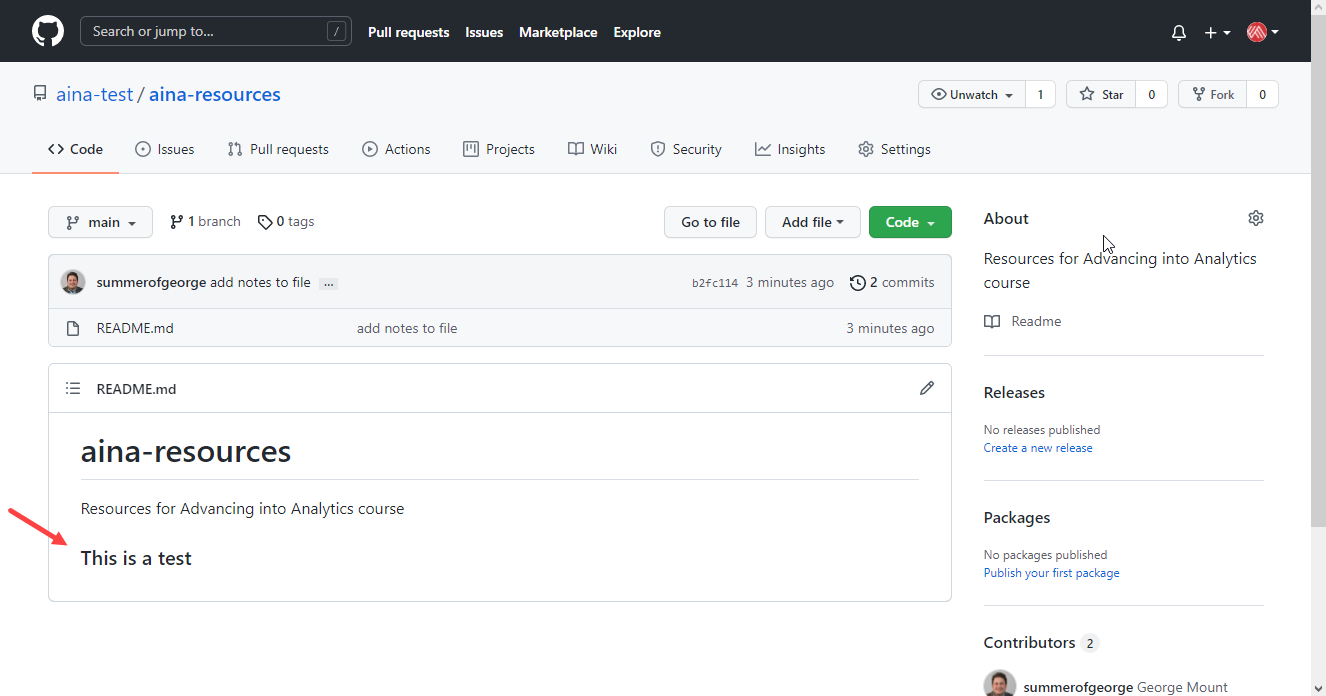
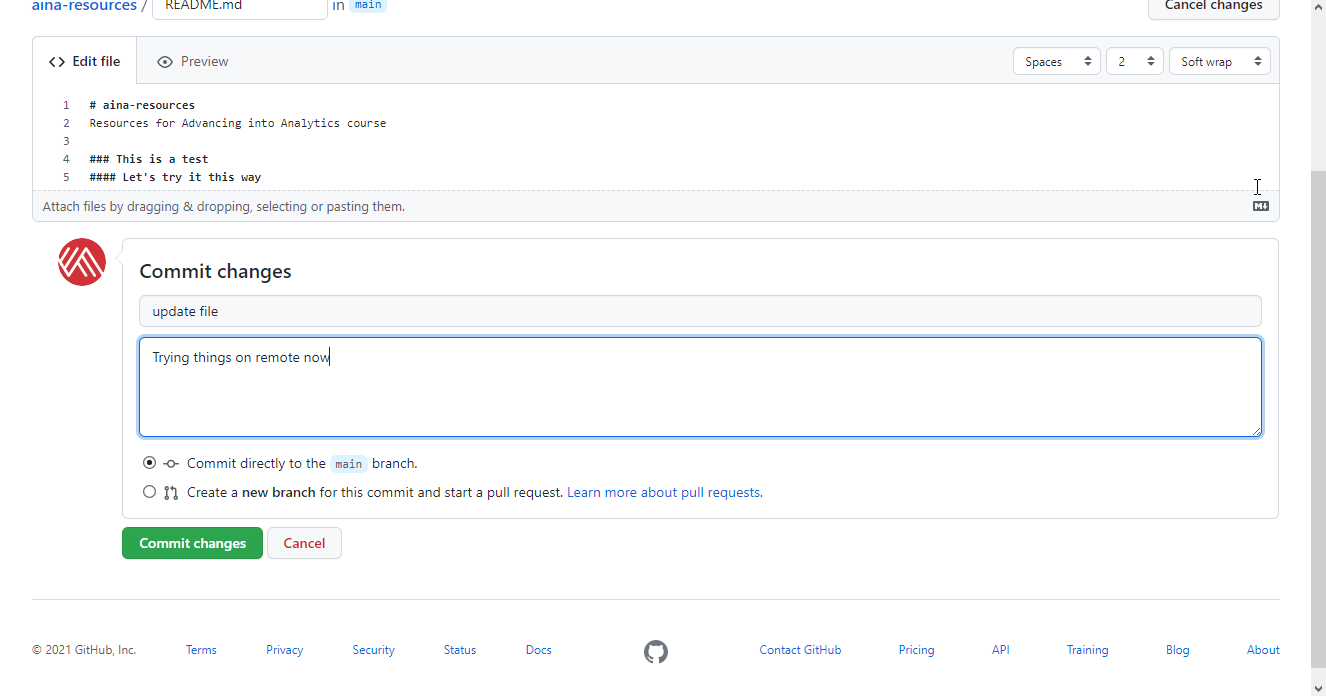
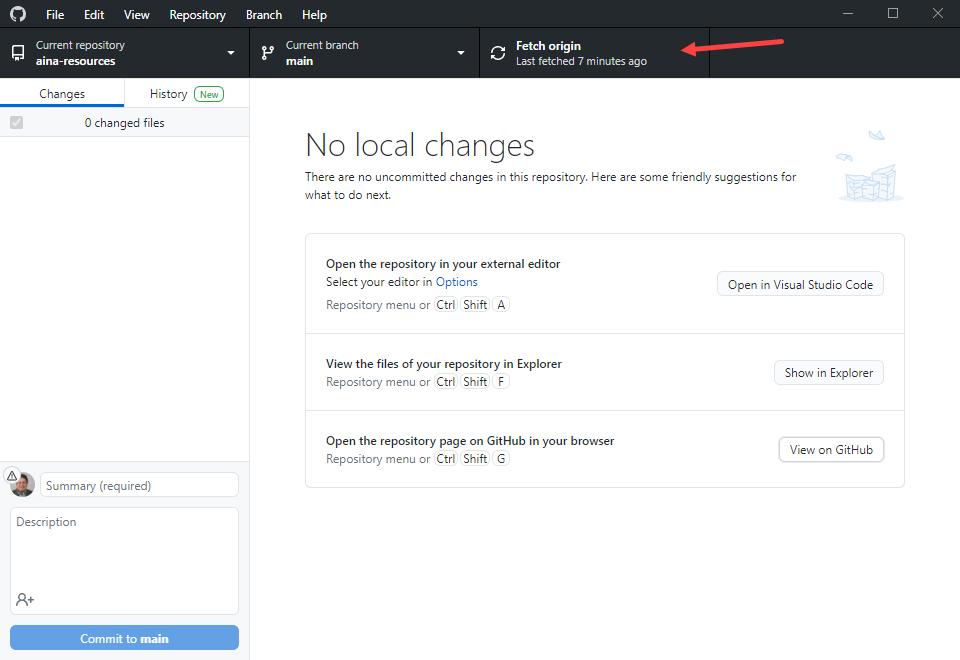
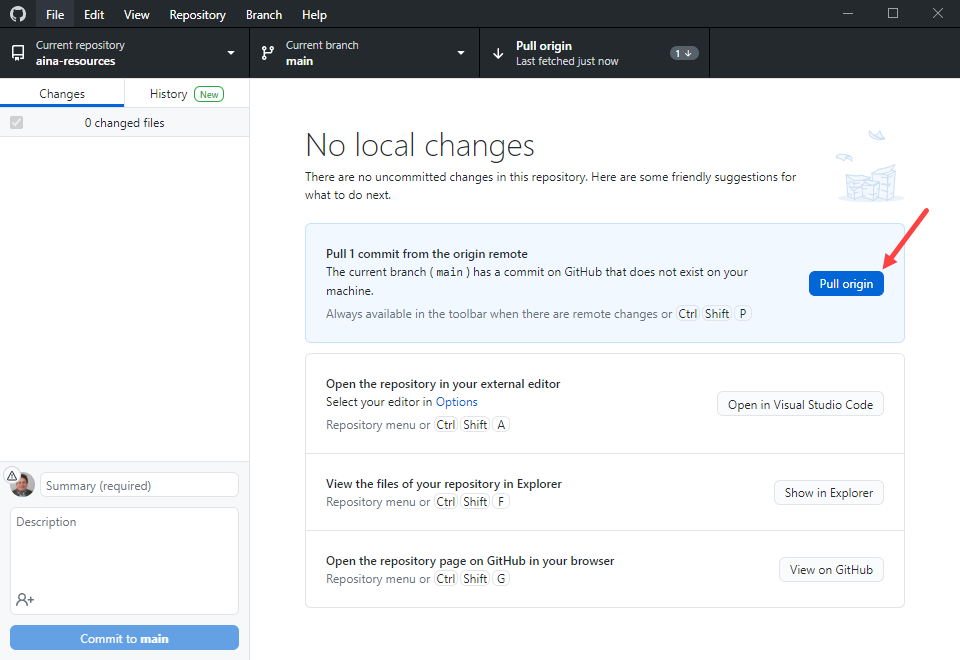
**Remote <> Origin via GitHub Desktop**

**Cloning a repo to GitHub Desktop**

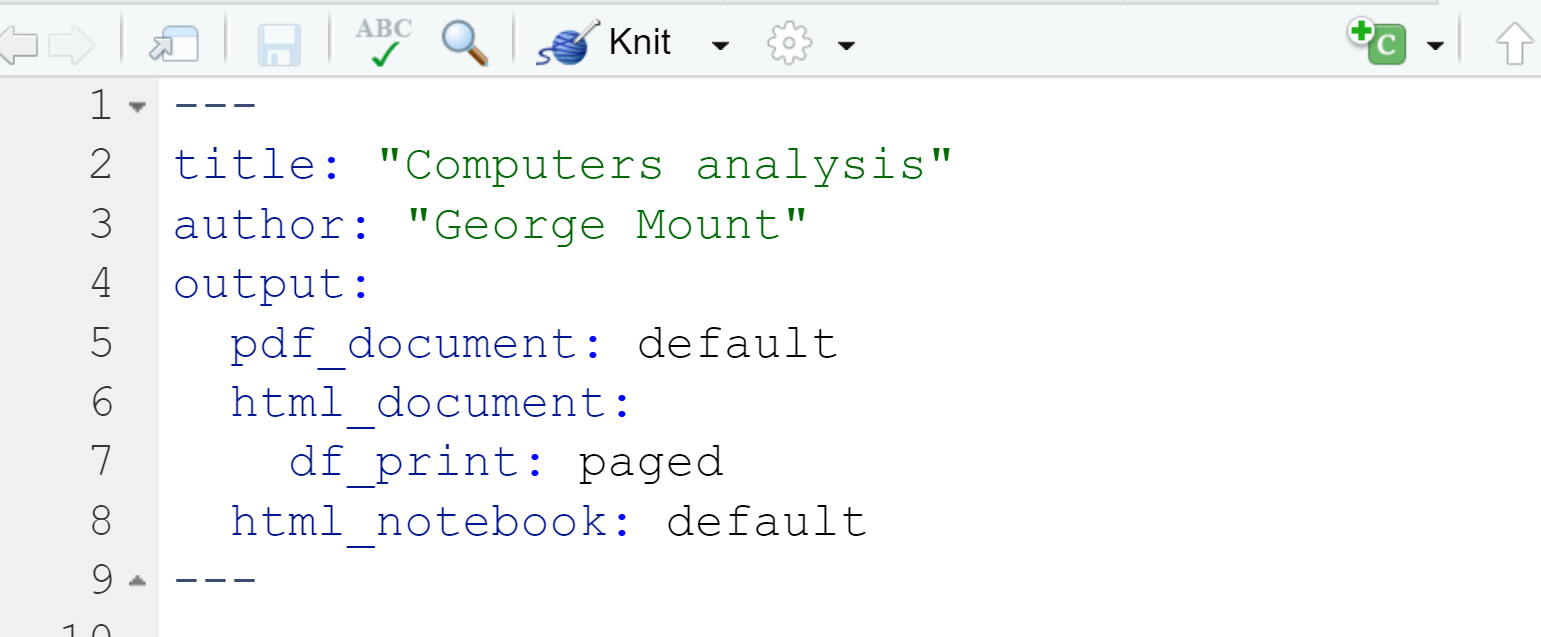
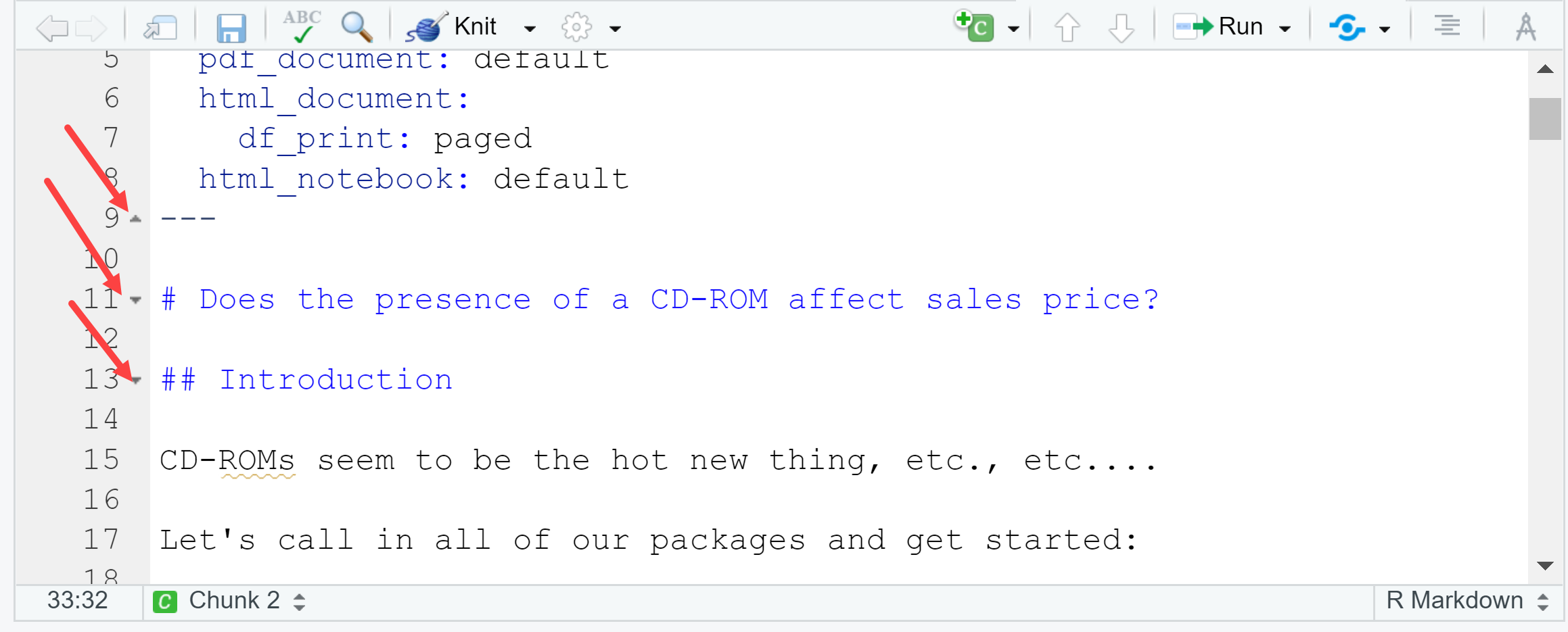
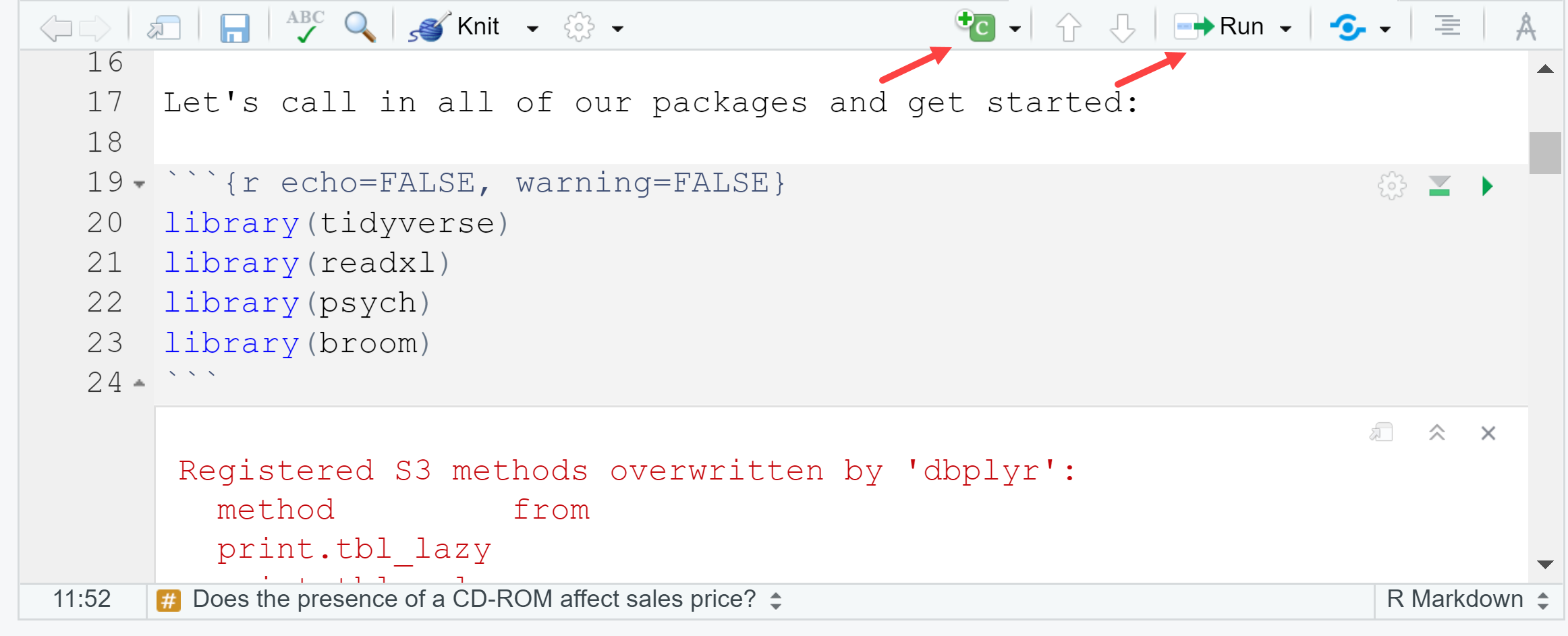
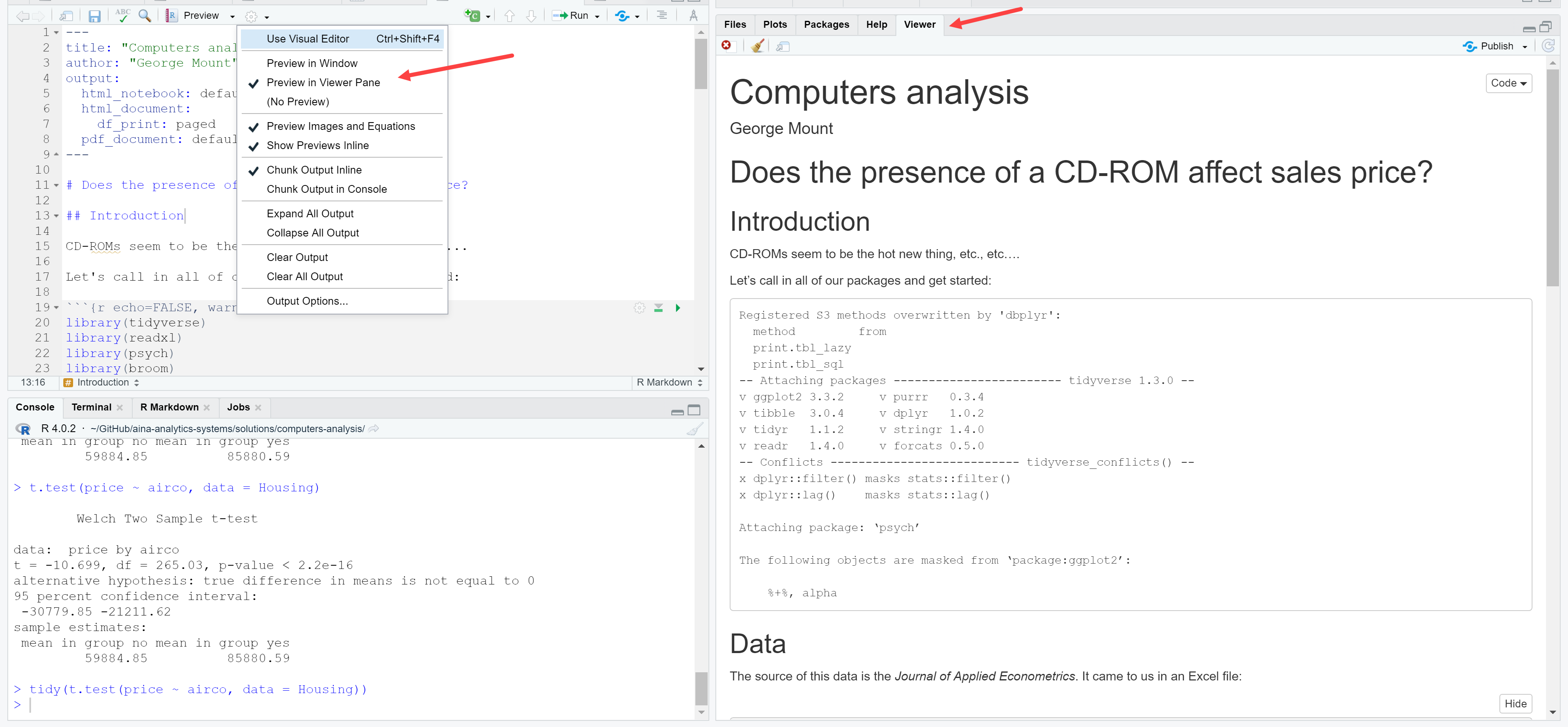
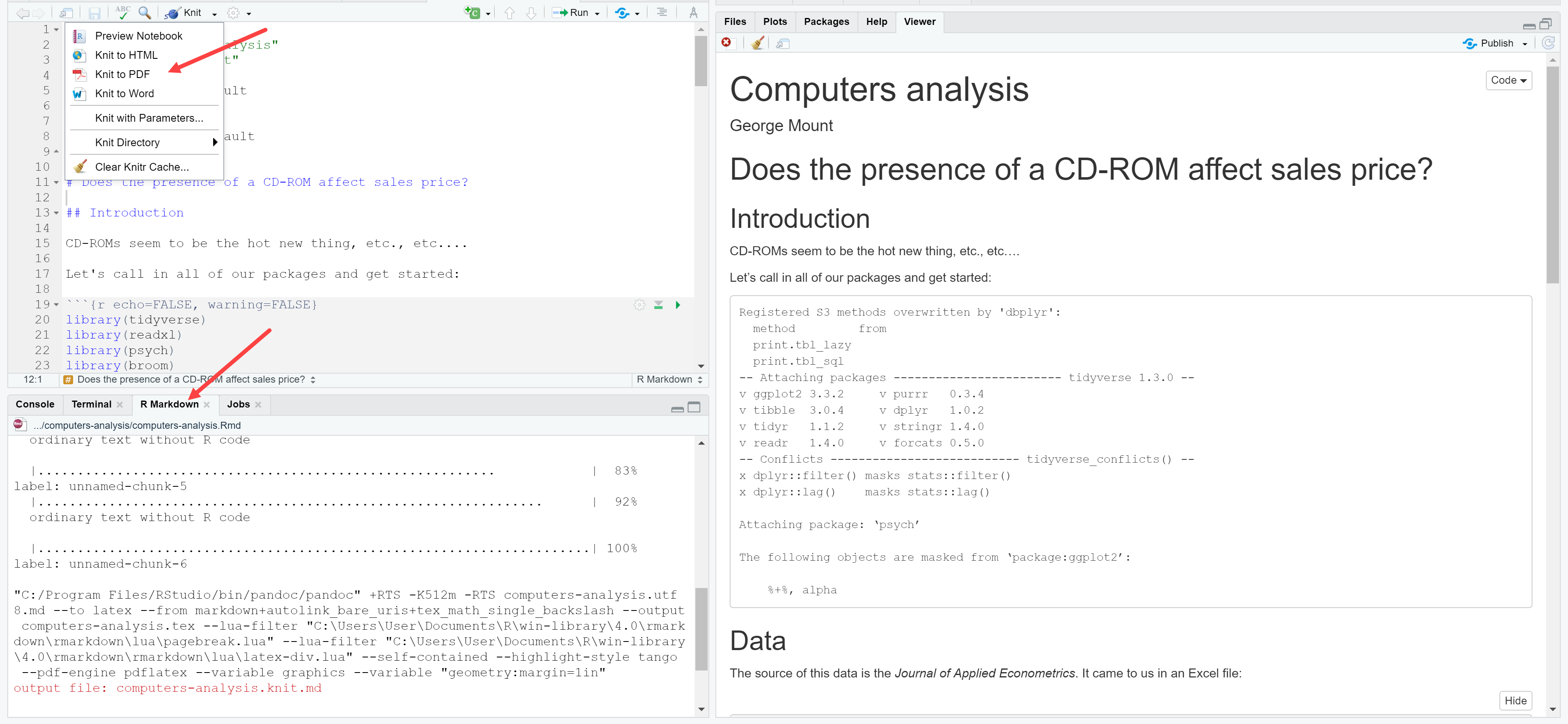
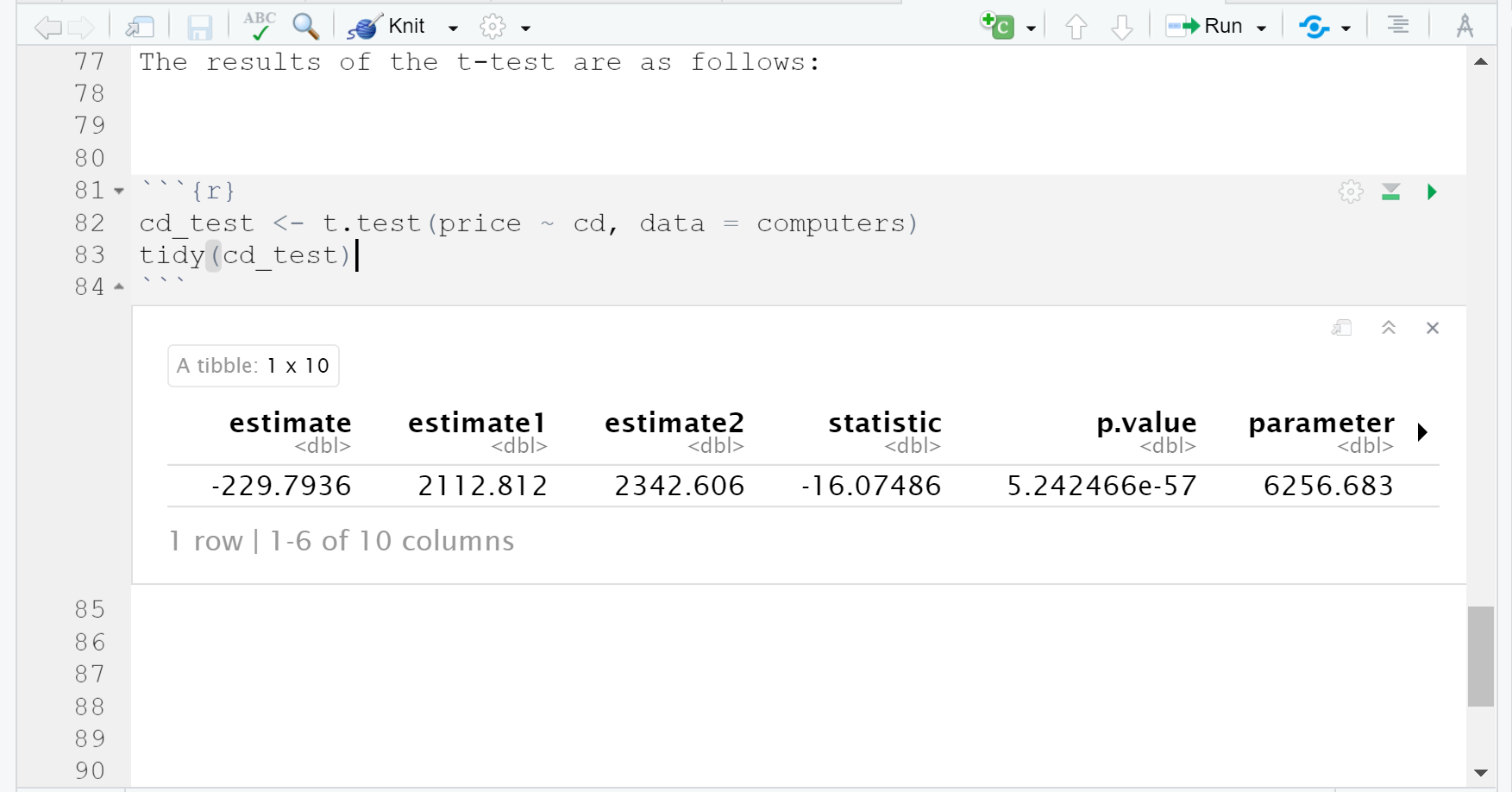
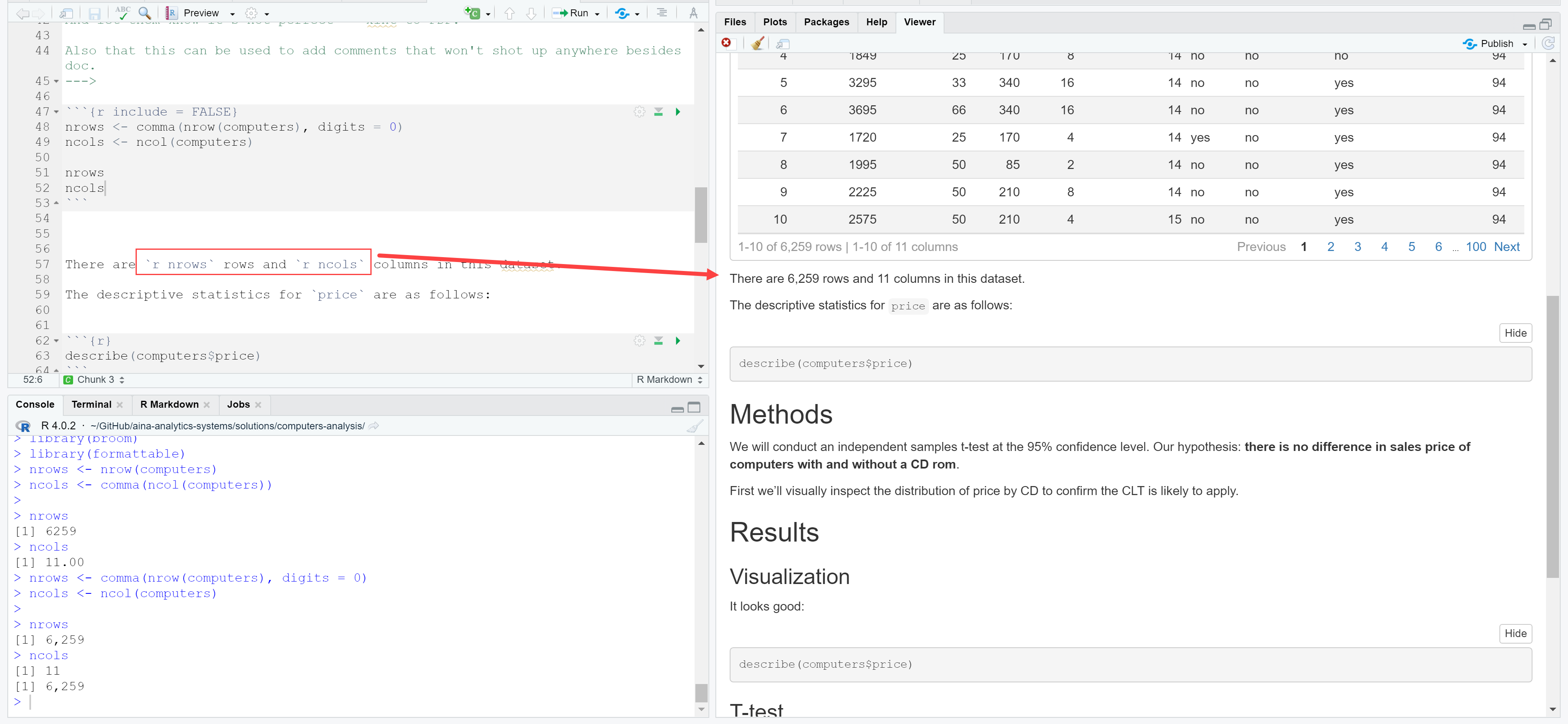
1. From the repo of interest, go to Code > Open with GitHub Desktop  
   
2. This dialog box will then open GH Desktop. Select what folder to clone this repo into.   
   
3. If you are forking a repo, it will ask you whether you’re using it to contribute to parent or for your own purposes. Choose the second.
4. You now have a copy of this repo *cloned* to your computer. It is *not* the source of truth – what lives on GitHub is the source of truth. To practice working on the so-called “origin,” or your local copy, let’s create a new repo.

**Creating a new repo**

So far we’ve been working with a *forked* repo; you could continue to do so using GitHub Desktop. But let’s create a brand-new repo instead.

1. Go back go to GitHub.com and select “New”   
   
2. Create the repo:
   1. Give the repo a name and say something about it.
   2. You can set it to private or public, your choice
   3. Add a readme file
   4. Select “Create repository”  
      
3. You can now clone the repo to your computer: Code > Open with GitHub Desktop
   1. Select where on your computer to store the repo
4. You’ve got a few options for how to access this repository’s data. You should see this in the middle of your Desktop session:   
   
5. Choose the second option to have a File Explorer session open up to the folder where this repo is located.
6. If your first option does not say “Open in Visual Studio Code,” go to File > Options > Integrations and select “Visual Studio Code” under External editor  
   
7. Click the option to “Navigate in Visual Studio Code.”
   1. Navigate the files toward the side to work with your files. VS Code can work with many files types, including Jupyter Notebooks.
      1. If you’d like to preview the Markdown inside of VS Code, Use Ctrl + K + V.
   2. Save your changes to the file, then head back to GitHub Desktop.   
      
8. You will see a list of changed files on the left, along with the contents of what’s changed to the right.
   1. Add a commit message and summary to the lower-left, then select “Commit to main.”  
      
9. You’ve committed or staged your changes to the main branch. You could continue committing locally if you’d like, but let’s go ahead and push this to origin.:  
   
10. Click on “View on GitHub” or otherwise get back to your repo to see the changes:   
    
11. Now let’s try this in reverse: make a change on GitHub.com and have it reflect on your computer. Go ahead and make some tweak to your readme file:  
    
12. Go back to your GitHub and select “Fetch origin” to check for any changes on GitHub.com (In this case, that’s the origin; you’re the remote!)  
    
13. Now click on “Pull Origin” (sound familiar)?  
    
14. You can now view these changes locally.

**R Notebooks: Helpful hints**

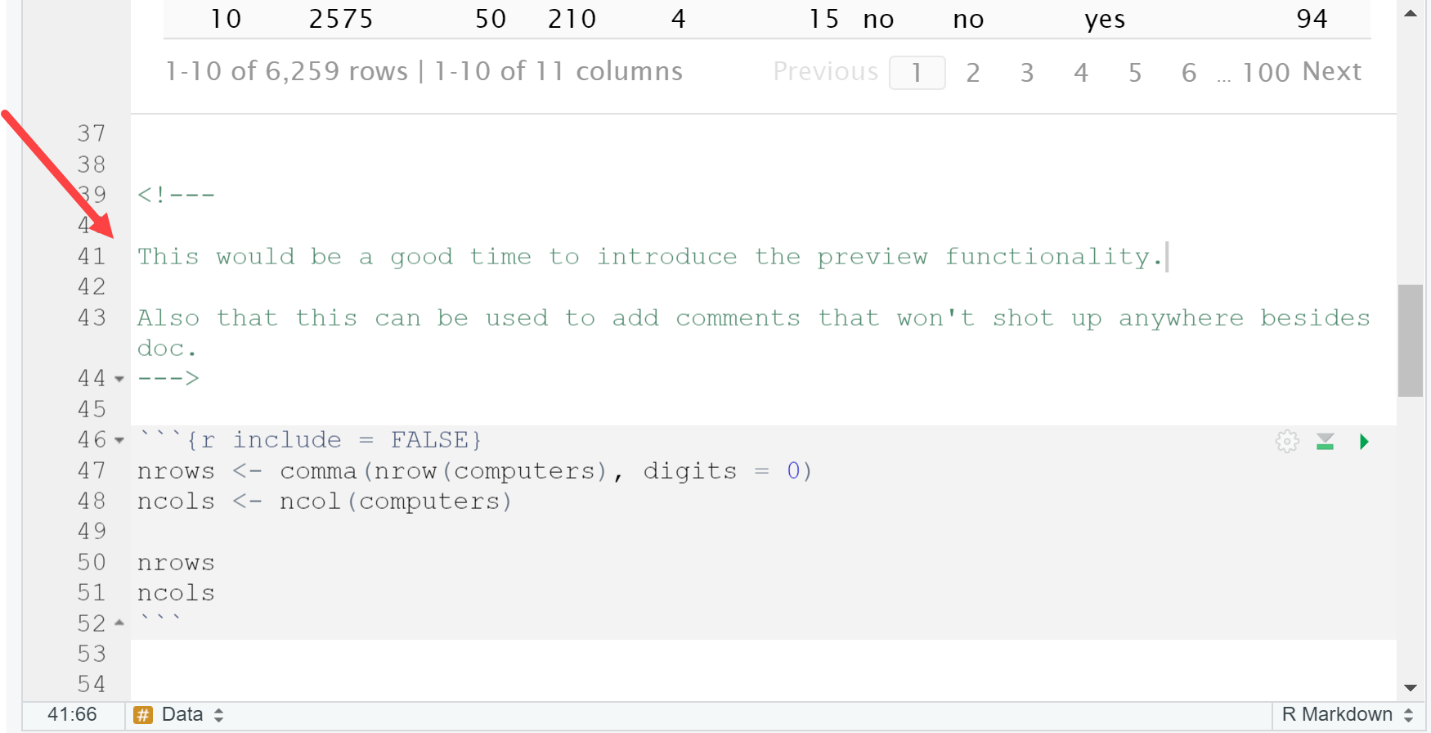
* The first part of the document is called the YAML (Yet Another Markup Language). This contains metadata about the file. You can give the document a title and author. As you export your file to different formats, those will display in the output section.  
  
* As you insert headers into the document, you will see button to the left to expand/contract the file:  
  
* The C and Run icons will let you, among other things, insert and run chunks, respectively:  
  
* You can customize the settings of chunks; we’ll use the following (as in previous screenshot). See more at the [official cheat sheet](https://rmarkdown.rstudio.com/lesson-15.html):
  + echo = FALSE will stop code from being shown in the output document
  + warning = FALSE will stop warnings from being shown in document
  + include = FALSE will run the chunk, but no code or output will be shown in the document.
  + message = FALSE will stop any messages being shown in the output document
* To view a preview of the document side-by-side with the R Notebook, click on the gearwheel of your document and select “Preview in Viewer Pane.” Then click “Preview” above your doc. The preview will show up in the Viewer pane.
  + This preview does not update on its own; to see subsequent changes to the doc, click Preview again or save the doc.   
    
* When you are ready to export the document, click under Preview > Knit to PDF (for example). You will see a log of the export process in the R Markdown tab. The document by default will be exported to wherever your R Notebook is.   
  
* The tidy() function can be used to present the result of many statistical tests in a more tabular format:  
  
* You can use in-line blocks for dynamically-rendered text using `r objectname` like so:  
  
* You can add text-only comments that aren’t rendered in the doc or run by R by using the following notation (comes from HTML):

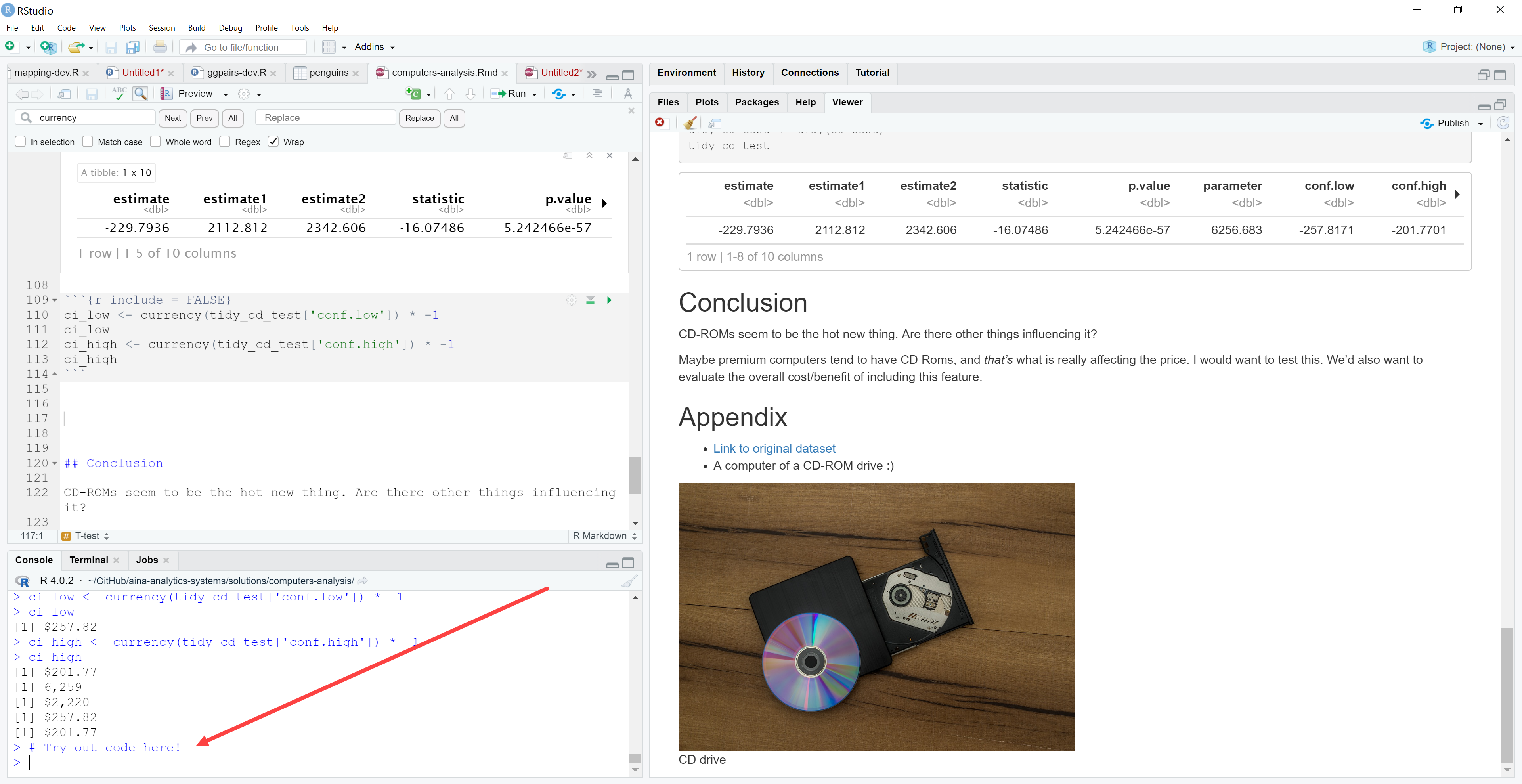
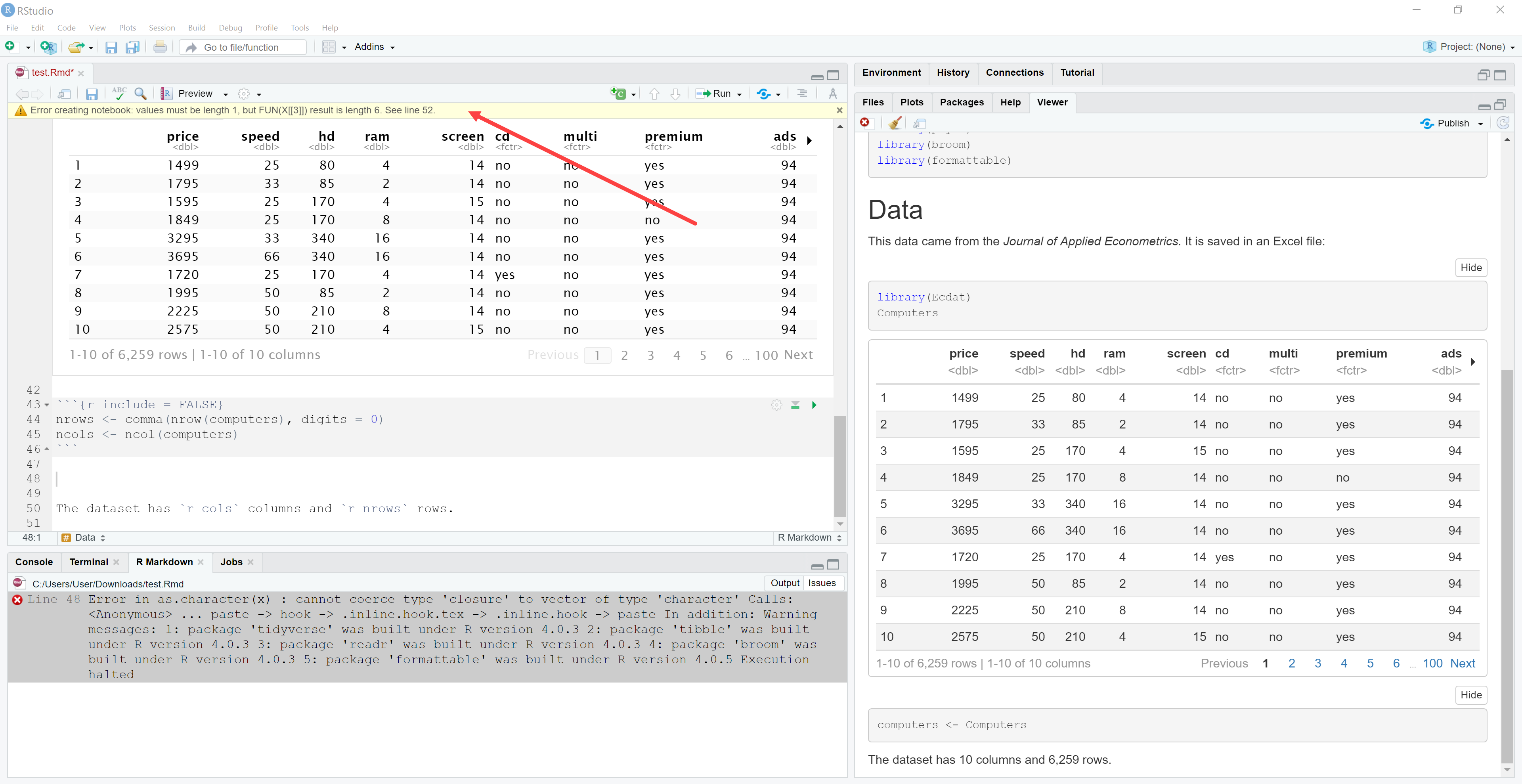
<!---

Comments go b/w the lines

--->

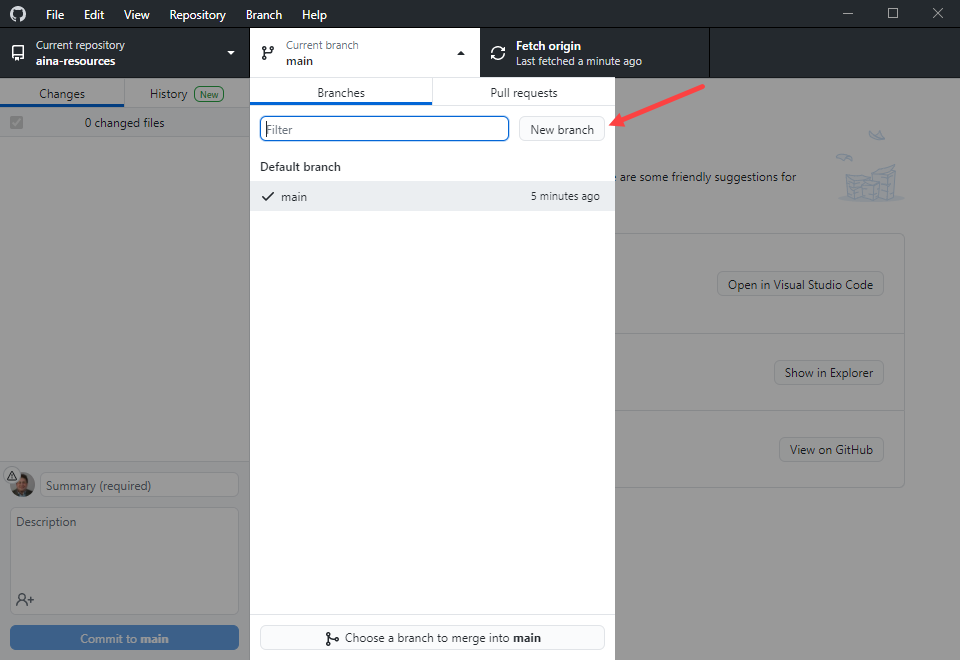
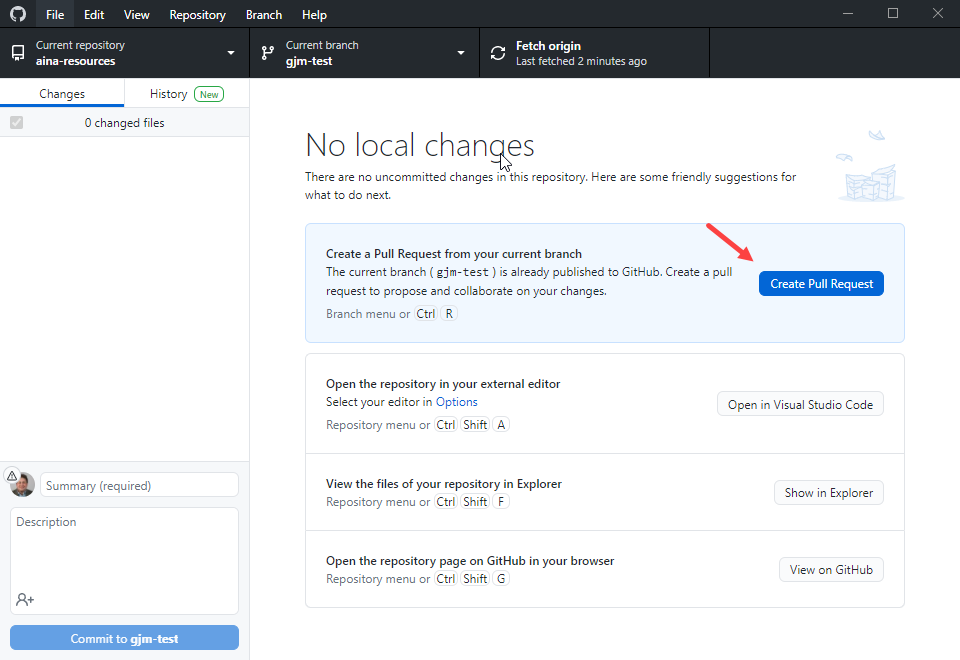
They’ll look like this in the notebook:



* If you’re just looking to try out some code before including it in the document, try using the console:  
  
* If you are having an issue with changes to your document not being shown in preview, there is likely an error with your notebook causing it not to build properly. Check the header of your doc for an error message or knit the file to get further details.  
  

**Working with branches**

You can work with branches locally via GitHub Desktop similar to on GitHub.com.

1. Click Current branch > New branch  
   
2. Give the branch a name > Create new branch > Create branch
3. If you click “Publish branch,” it will be made available on GitHub.com
4. Make any changes on the branch using VS Code, etc.
   1. You will commit and push your changes like before, then go to GitHub to open and manage pull requests.
5. After you’ve made the changes GitHub Desktop will ask if you want to Create Pull Request. Go ahead and click it:  
   
6. This will bring you back to GitHub.com where you’ll walk through the PR steps again.