

Trackdown Demo

Neil J. Hatfield

Nov. 4, 2022

Purpose

This file is to provide us with an opportunity to learn about the **trackdown** package, see some examples of the commands from this package, and actually get to use the package.

The **trackdown** package provides a way to work with collaborators on the narrative portions of your documents. This package works by uploading your RMD file to a Google drive and making type of Google Doc. You can then share that Google Doc with your collaborators and take advantage of the editing tools of that platform. Once you and your co-authors have finished editing (and have accepted all changes), you can use **trackdown** to pull those edits out of the Google Doc and back into your RMD file.

Some Important Notes

The Google Doc that gets created isn't quite like a normal Google Doc, nor is it a copy of the RMD file. For instance, you can set all of your code chunks to *not* appear in the Google Doc. This is a good practice to do since editing code chunks in Google Docs can cause issues. There will be placeholders for code chunks and other elements that aren't part of the narrative text (e.g., YAML header).

Additionally, the output of your code chunks won't appear in the Google Doc. Thus, I recommend that you take advantage of **trackdown**'s ability to simultaneously upload a knitted version of your RMD. The knitted file will be a PDF added to your Google Drive, thus you can share this file with your collaborators so they can see the outputs. Using section headings and and temporary text can be a great way to help create connection points between the Google Doc and the initial PDF.

Installing Trackdown

To make use of **trackdown**, you will need to first install this package (and any dependencies). While there is a version available on CRAN, the development version is fairly stable and has some **powerful** features. You will need either **devtools** or **remotes** installed to install the development version.

```
devtools::install_github(  
  repo = "claudiozandonella/trackdown",  
  build_vignettes = TRUE  
)
```

If using remotes, just replace "devtools" with "remotes" (no quotation marks).

Working with `trackdown`

There are four main functions in `trackdown`

- 1) `upload_file`
- 2) `update_file`
- 3) `download_file`
- 4) `render_file`

Important Caution

IF you are going to put any of these commands into your RMD file, place them into their own code chunks and be sure to set `eval = FALSE` in the chunk options. (You might also want to set `include = FALSE` as well.) The commands here in `trackdown` run better from your console rather than in chunks, especially if you need to log into your Google drive.

Uploading the RMD

The first function that you'll need use is the `upload_file` command. This is what will send your RMD file to your Google drive. The first time you use `trackdown` (or if your prior login has expired), you'll be prompted to log into your Google Drive. If you have several connected Google accounts, you'll be asked which one to send the file to. This interactivity is why you should not run this code in a code chunk nor evaluate this code when knitting your RMD.

You will need to specify the following arguments:

- The RMD file—provide the file name AND `.Rmd`. If your file is not currently in your working directory (look at the bar just above the console), you'll need to provide the full file path.
- The output file—provide the file name AND either `.pdf` or `.html` (the HTML file will be converted to a PDF). Again, if the file is not in your current working directory, you'll need to give the full file path.
 - Be sure to knit your RMD first so that an output file exists!
- Use `hide_code` and `rich_text` arguments, both set to `TRUE`. These arguments hide the code in the Google Doc and highlight the portions that you and your collaborators shouldn't mess with while in Google Docs.
- There are some additional/optional arguments that you can explore on your own.

The following code provides an example of how to use `upload_file`:

```
trackdown::upload_file(  
  file = "TrackdownDemo.Rmd",  
  hide_code = TRUE,  
  rich_text = TRUE,  
  path_output = "TrackdownDemo.pdf"  
)
```

By default, `trackdown` will create a new folder in your Google Drive called “trackdown” and add all files there. (This is an option you can change if you wish.) The output file will have “-output” appended to the end of the file name in your Google drive.

Updating the Google Doc

Let's suppose that after you've uploaded your RMD file to your Google drive, you make some edits and want add those so your collaborators can see them. Instead of using `upload_file`, you'll use `update_file`.

The arguments to `update_file` are the same as for `upload_file`, the major difference is that `update_file` is set to look for the existing information about where to put your changes while `upload_file` assumes the file doesn't exist in your Google Drive. If you use the the wrong command, not the end of the world. You'll get a warning message.

Here's an example of using `update_file`:

```
trackdown::update_file(  
  file = "TrackdownDemo.Rmd",  
  hide_code = TRUE,  
  rich_text = TRUE,  
  path_output = "TrackdownDemo.pdf"  
)
```

Again, be sure to knit your RMD file so that your updates get added to the output file as well.

Downloading the Changes

Once you and your collaborators are finished OR you've reached a point where you want to pull the changes and update the output file, you'll use the function `download_file`.

The most important argument is `file`. This needs to be the file path to the local RMD file. Again, if the RMD is in your current working directory you can give just `fileName.Rmd`. But if the local RMD file is *not* in your working directory, you'll need to give the full file path.

Here's an example of using `download_file`:

```
trackdown::download_file(  
  file = "TrackdownDemo.Rmd"  
)
```

You will want to be careful with the `download_file` command as this WILL replace your local copy of the RMD file. Once you've downloaded the changes, you can work with the RMD file locally. You can make further changes, knit the file, and even update the Google Doc version via `upload_file`.

There are additional arguments in this function for navigating the Google Drive if you have moved the file, renamed the file, or renamed folders.

Render The File

The last function you can use is `render_file`. This command is essentially the `download_file` command AND immediately knits the RMD file, producing an updated knitted output file(s).

Again, the only necessary argument is `file` which acts the same as in the `download_file` function.

Here's an example of using `render_file`:

```
trackdown::render_file(  
  file = "TrackdownDemo.Rmd"  
)
```

DELETE ME AS PART OF A DEMO—The knitting of your document happens automatically and silently—Viewer window will not open.

Example Time!

Let's try an example of collaboration. I'm going to send out a link to this RMD file in my Google Drive. I would like each of you to add your name to the following list. Be sure to include a plus sign (+) followed by a space before your name. Follow your name with a set of parentheses and put your Collaboration Partner's name inside them.

Each person should make their own line. When finished feel free to look at the rest of this file as well as the shared output file.

- Neil Hatfield (The Entire Class)
- Leigh Stern (Katie Clayton)
- Katie Clayton (Sarah Kircher)
- Sarah Kircher (Angel Perez)
- Grace Wagner (Marco)
- Eric Wu (Grace Wagner)
- Nathanael Nemias (Aiden)
- Aiden (Gaurang)
- Veronica Jenkins (Xiao Lei)
- Xiao Lei (Rui Du)
- Rui Du (Sagan Kakkar)
- Arushi Singh (Alex P)
- Ian Munoz (Veronica)
- Hongyu Guo (Sagan)
- Evan Settipane (Anika Deshpande)
- Ethan Sullivan (Nate)
- Alex Piechucki (Arushi)
- Adith Gopal (Leigh Stern)
- Angel Perez (Eric W)
- Sagan Kakkar (Hongyu Guo)
- Nathanael Kovscek (Adith Gopal)
- Matteo Martone (Jack Billings)
- Olachi Mbakwe (Junki (Dean) Bae)
- Junki (Dean) Bae (Olachi Mbakwe)
- Gaurang Kakade (Ian)
- Anika Deshpande (Evan Settipane)
- Jack Billings (Arushi Singh)
- Arushi Singh (Marco)
- Jingcheng Xiao (Nathanael Nemias)
- Owen Braglio (Nithika Radhakrishnan)
- Nithika Radhakrishnan (Owen Braglio)

A Second Example

Here's an example of how you might post a visualization in the output file and leave a tag location for your co-authors to write narrative text about the plot.

[Text here for Figure 1]

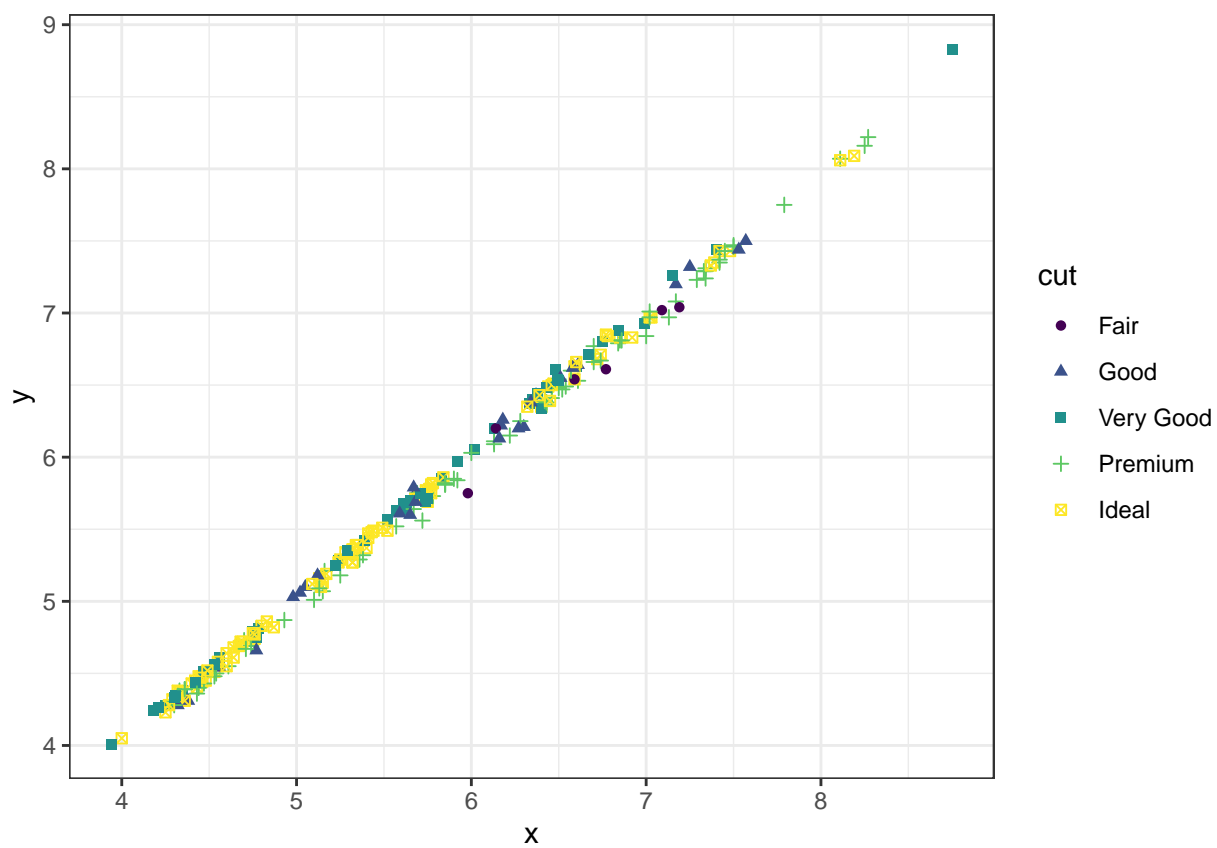


Figure 1: Diamonds Plot

Code Appendix

```
knitr::opts_chunk$set(
  echo = FALSE,
  warning = FALSE,
  message = FALSE,
  fig.align = "center"
)

# Load packages ----
library(trackdown)
library(dplyr)
library(ggplot2)

devtools::install_github(
  repo = "claudiozandonella/trackdown",
  build_vignettes = TRUE
)

# If using remotes, just replace "devtools" with "remotes" (no quotation marks).

trackdown::upload_file(
  file = "TrackdownDemo.Rmd",
  hide_code = TRUE,
  rich_text = TRUE,
  path_output = "TrackdownDemo.pdf"
)

trackdown::update_file(
  file = "TrackdownDemo.Rmd",
  hide_code = TRUE,
  rich_text = TRUE,
  path_output = "TrackdownDemo.pdf"
)

trackdown::download_file(
  file = "TrackdownDemo.Rmd"
)

trackdown::render_file(
  file = "TrackdownDemo.Rmd"
)

set.seed(184)
diamonds %>%
  slice_sample(n = 250) %>%
  ggplot(
    mapping = aes(x = x, y = y, color = cut, shape = cut)
  ) +
  geom_point() +
  theme_bw()
```