

INTRODUCTION TO R MARKDOWN AND LITERATE PROGRAMMING

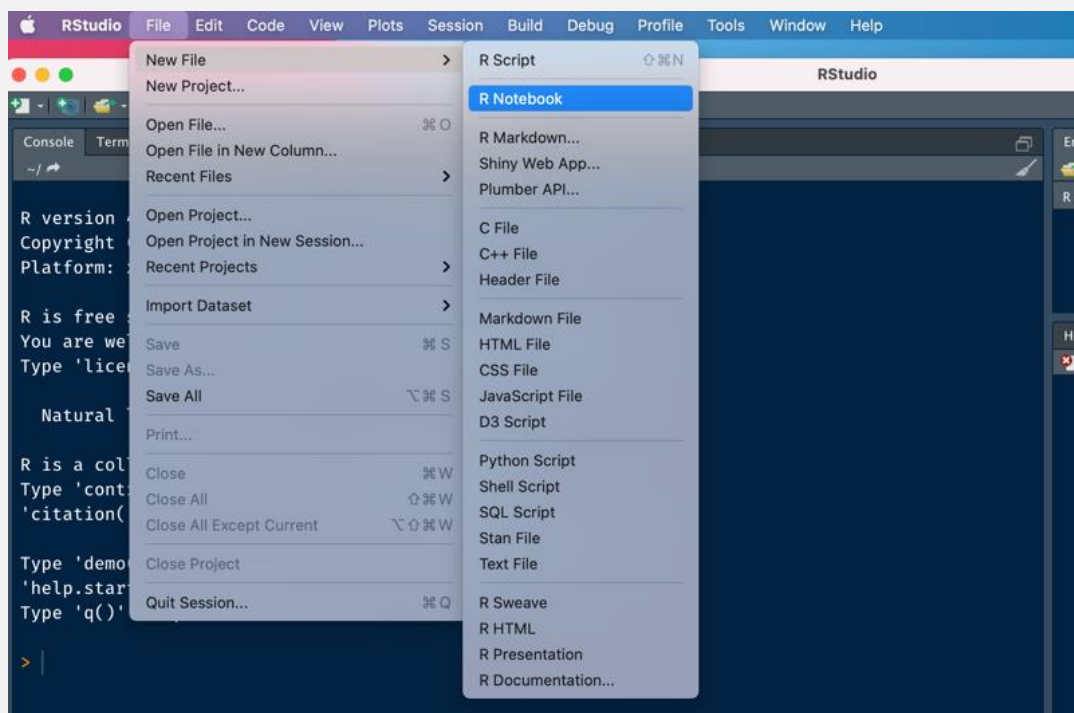
"R Markdown documents are fully reproducible. Use a productive [notebook interface](#) to **weave together narrative text and code** to produce elegantly formatted output. Use [multiple languages](#) including R, Python, and SQL." --- [R Markdown website](#)

In this post, I show how to get started with R Markdown and "literate programming" to create dynamic documents that contain both text and code. These can be run interactively, as you go, and to generate high quality documents or reports. I attach an example R Notebook with this post that includes some comments and set-up notes and resources.

Open your first R Notebook

If you're new to R and R Studio, then you can install R using at <https://www.r-project.org> and R Studio at <https://rstudio.com>.

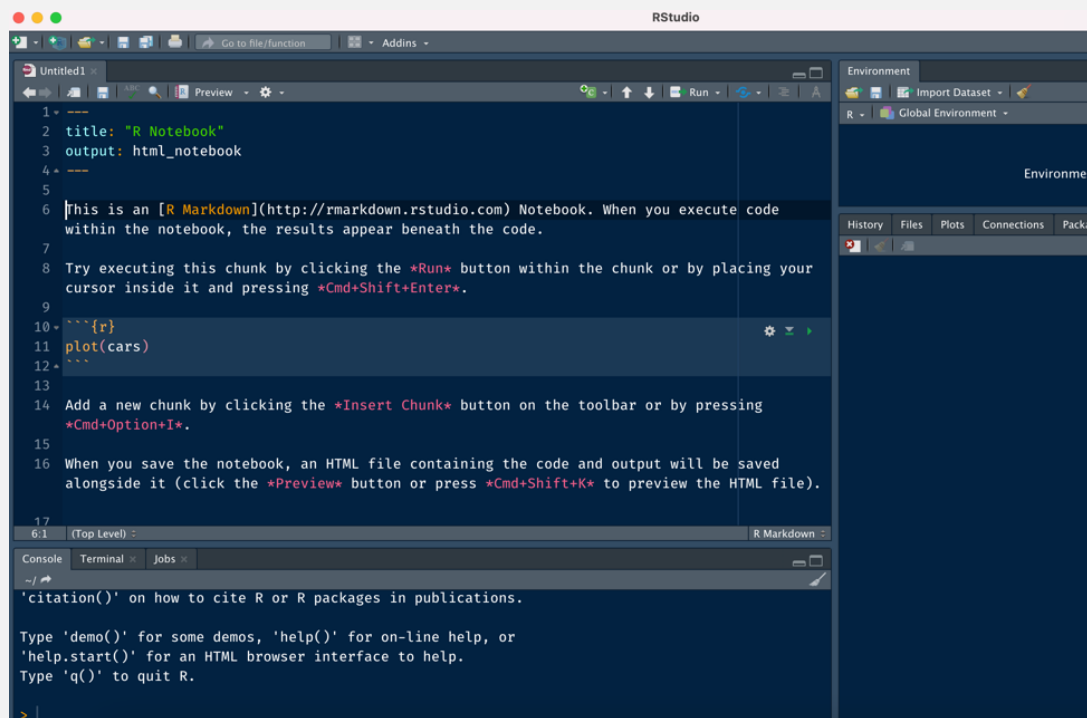
To start an R Notebook, open R Studio, and then go to *File > New File > R Notebook*



An [R Notebook](#) is "...a document with chunks that can be executed independently and interactively, with output visible immediately beneath the input." Your document can be interacted with, as you write, i.e. you can tweak your code, or update your data source. R Markdown uses [Markdown](#) syntax for formatting text: "...is a lightweight markup language for creating formatted text using a plain-text editor." (Wikipedia)

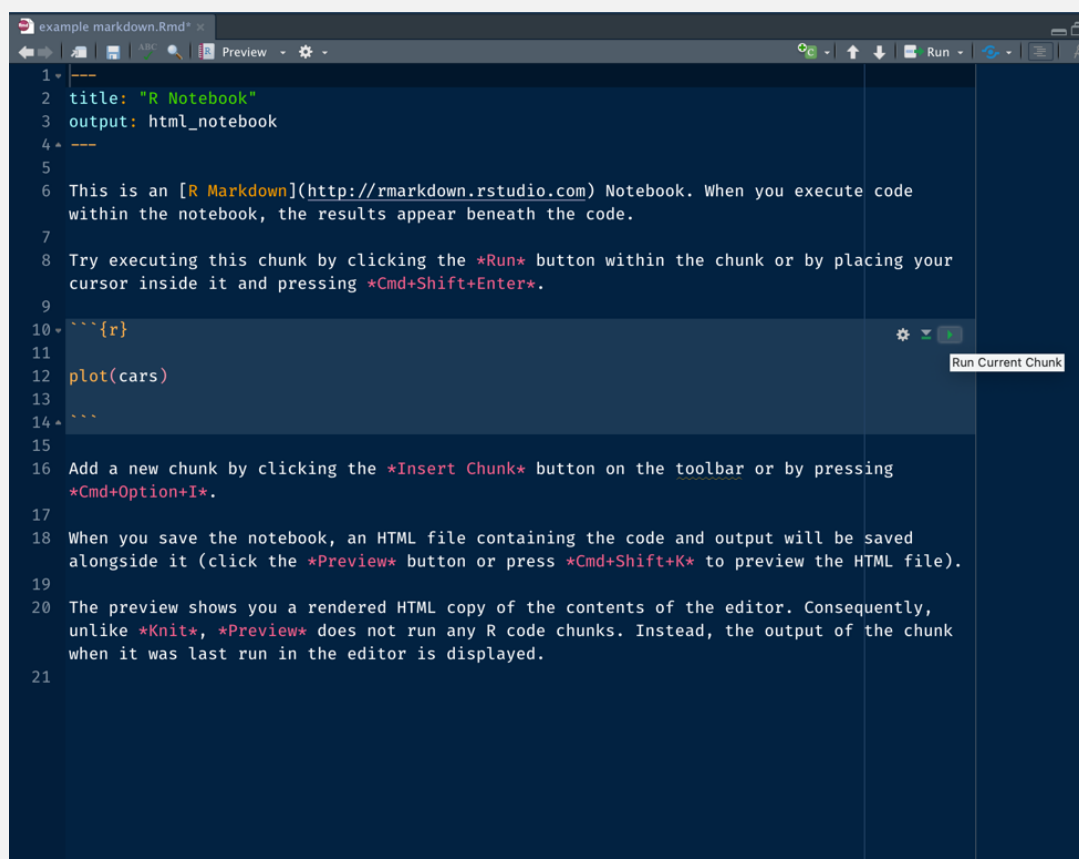
We use the "[literate programming](#)" idea, where one dynamically weaves together narrative writing and executable code (and code output) in the same environment.

When you open an R Markdown document, it already includes a quick example with text, and an R code chunk. See below:

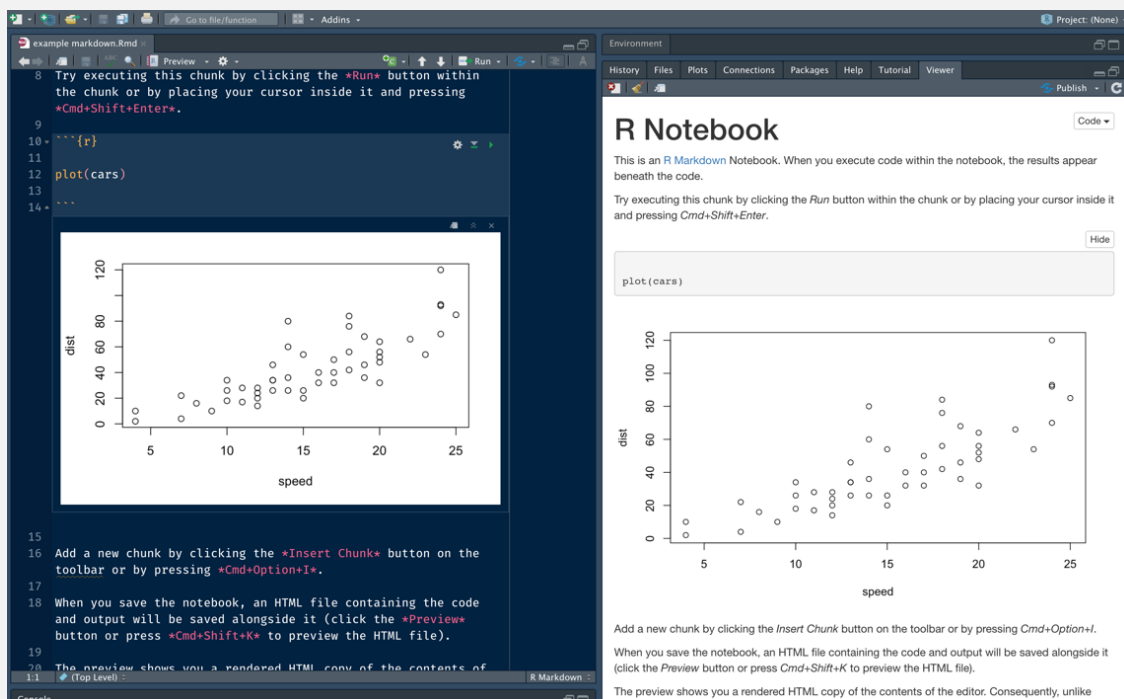


In R Markdown one uses code “chunks”, which are little blocks for code inside your document that you can intersperse, code in, customize, and run directly in your document (click the “play button” in the chunk and see the example plot render in your notebook.)

As shown in the image below, we can also run the code separately instead of running the whole file:



The (HTML) document on the right is generated once we run the file above (i.e. when you click “Preview”). One can also “knit” to PDF or Microsoft Word.



Additional features of R markdown include-

- You can customize the notebook to use different languages (R, Python, and SQL) --- so you can have a single document where you incorporate various different coding languages and data sources in the same place.
- You can also include **in-line code**, e.g. “*The mean age is ``r mean(data$age)`` years*”
- You can convert it to various output formats like HTML, PDF, or Microsoft Word for sharing. (In fact, there are many ways to share your R Markdown documents --- see some of the resources below.)

R Markdown uses the Markdown syntax for formatting text. Markdown is a very simple way to format plain text into heading levels, paragraphs, bullet points, lists, and one can also incorporate LaTeX for technical writing.

Some Applications

- Weaving together code and writing means you can create "dynamic" documents that update as more data comes in or your inputs change.
 - This is great for things like High Frequency Checks, or data validation reports.
- These documents can be highly customized.
 - For instance, if you’re outputting an HTML document, you can incorporate custom CSS.
- And because you’re using macros and chunks, and this is all code, you can also programmatically manipulate these documents.
- Other uses include: Technical papers, technical sketches, academic papers, dashboards, webpages, and “control” documents with various parameters (a meta document you can use to control various scripts all in one place).

Resources

- [R Markdown: The Definitive Guide](#) --- Yihui Xie, J. J. Allaire, Garrett Grolemund
- [R Markdown Cookbook](#) --- Yihui Xie, Christophe Dervieux, Emily Riederer
- [Meta RMarkdown - Taxonomy and Use cases](#) --- "A meta collection of all things R Markdown", from the Mockup Blog, Thomas Mock
- [RMarkdown Driven Development \(RmdDD\)](#) --- Emily Riederer
 - See also [Column Names as Contracts](#) (on data validation with {pointblank})
 - For Markdown, see: <https://www.markdownguide.org/>
- [bookdown](#) --- "Write HTML, PDF, ePub, and Kindle books with R Markdown"
- [blogdown: Creating Websites with R Markdown](#) --- Yihui Xie, Amber Thomas, Alison Presmanes Hill
- [Distill for R Markdown Scientific and technical writing, native to the web](#) --- "Distill for R Markdown is a web publishing format optimized for scientific and technical communication"
- [prettydoc](#) --- HTML themes for R Markdown
- [Advanced R Markdown](#) --- "Day 2: Customization and Extensions, 2019/01/16 @ rstudio::conf, Austin, TX", Yihui Xie and Hao Zhu
- [{knitr}](#) --- "The R package **knitr** is a general-purpose literate programming engine, with lightweight API's designed to give users full control of the output without heavy coding work"
- [{gluedown}](#) --- "The goal of {gluedown} is to ease the transition from R's powerful vectors to formatted markdown text."
- [Lists are my secret weapon for reporting stats with knitr: Tidying and splitting model summaries for inline reporting](#) --- Tristan Mahr's blog
- Adding citations to your documents:
 - [Adding citations to posts - A minimal example website using blogdown](#)
 - [Bibliographies and Citations in R Markdown](#)
 - [GitHub - crsh/citr: RStudio Addin to Insert Markdown Citations](#)
 - [Better BibTeX for Zotero](#)
 - [Primer — An Introduction to CSL — Citation Style Language 1.0.1-dev documentation](#)
 - [knitcitations](#)