

The background of the slide is a complex, abstract network diagram. It consists of numerous nodes of varying sizes and colors (dark blue, light blue, and grey) connected by a dense web of thin, light grey lines. Some nodes are highlighted with larger, concentric circles. The overall aesthetic is technical and interconnected.

REPRODUCIBILITY WITH THE RMARKDOWN/KNITR ECOSYSTEM

Melinda Higgins, PhD; Emory University, Professor
Dated 25 March 2021

Think about your own work...

- What do you want to automate?
- What could you re-use?
 - code, files, formatting, graphics, logos, header, footer, boilerplate
- What should you share with your team?
- What do you find yourself doing over and over?
 - correcting or reformatting
- If you won the lottery today (and left your job), what do you need to tell your replacement so they can pick up where you left off and complete your current tasks?

THE BIG PICTURE

data

text

code

figures

tables

- Manuscript
- Report
- Slides
- Website
- Dashboard
- Book

A (BRIEF) HISTORY OF LITERATE PROGRAMMING

Literate Programming > Dynamic Documentation
> [R]Markdown

YEAR	Event
1992	"Literate Programming" is introduced by Donald Knuth as "that (which) combines a programming language with a documentation language, thereby making programs more robust, more portable, more easily maintained, and arguably more fun to write than programs that are written only in a high-level language. The main idea is to treat a program as a piece of literature, addressed to human beings rather than to a computer. " http://www-cs-faculty.stanford.edu/~knuth/lp.html
2002	Friedrich Leisch introduces SWEAVE a program for "Dynamic generation of statistical reports using literate data analysis" https://leisch.userweb.mwn.de/Sweave/

Literate Programming > Dynamic Documentation

> [R]Markdown

YEAR	Event
2004	John Gruber created the Markdown language in 2004 in collaboration with Aaron Swartz - their goal was to "write using an easy-to-read, easy-to-write plain text format, and optionally convert it to structurally valid XHTML (or HTML)" https://daringfireball.net/projects/markdown/
2012	Yihui Xie releases <code>knitr</code> R package released - <code>knitr</code> was inspired by SWEAVE
2014	<code>rmarkdown</code> R package released - extends Markdown to work with R/RStudio environment

SWEAVE by Friedrich Leisch

Sweave: Dynamic Generation of Statistical Reports Using Literate Data Analysis

What is Sweave?

*“Sweave is a tool that allows to embed the R code for complete data analyses in latex documents. The purpose is to create dynamic reports, which can be updated automatically if data or analysis change. Instead of inserting a prefabricated graph or table into the report, the master document contains the R code necessary to obtain it. When run through R, all data analysis output (tables, graphs, etc.) is **created on the fly and inserted** into a final latex document. The report can be **automatically updated if data or analysis change, which allows for truly reproducible research.**”¹³*

13. Friedrich Leisch. Sweave: Dynamic generation of statistical reports using literate data analysis. In Wolfgang Härdle and Bernd Rönz, editors, Compstat 2002 - Proceedings in Computational Statistics, pages 575-580. Physica Verlag, Heidelberg, 2002. ISBN 3-7908-1517-9.

The next evolution <- knitr



In 2012 Yihui Xie, created and released the `knitr` package for R to extend the capabilities of SWEAVE beyond LaTeX.

*"The `knitr` package was designed to be a transparent engine for dynamic report generation with R, solve some long-standing problems in Sweave, and combine features in other add-on packages into one package."*¹⁴

¹⁴: <https://yihui.org/knitr/>

The next evolution <- ... + rmarkdown



- In 2014, RStudio released rmarkdown to extend the markdown language originally intended to write documents for the “web” (*i.e. HTML*).¹⁵
- rmarkdown leverages **Pandoc** (“universal document converter”) ¹⁶ to convert between formats: from HTML (readable by web browsers) to DOC (such as from Microsoft Word or Google Docs) to ODT (Libre Office) to PDF (portable document format) to others like EPUB (e-books), HTML5 slide shows (slidy, ioslides), and TeX based documents and slides (Beamer).

15. <https://daringfireball.net/projects/markdown/syntax>

16. <http://pandoc.org/index.html>

Pandoc <https://pandoc.org/>

...often called the *Swiss-Army knife* for converting files from one format to another. Pandoc can convert documents in markdown, reStructuredText, textile, HTML, DocBook, LaTeX, MediaWiki markup, TWiki markup, OPML, Emacs Org-Mode, Txt2Tags, Microsoft Word docx, LibreOffice ODT, EPUB, or Haddock markup to

- HTML formats: XHTML, HTML5, Slidy, reveal.js, Slideous, S5, DZSlides.
- Word processor formats: Microsoft Word docx, OpenOffice/LibreOffice ODT, OpenDocument XML
- Ebooks: EPUB version 2 or 3, FictionBook2
- Documentation formats: DocBook, TEI Simple, GNU TexInfo, Groff man pages, Haddock markup
- Page layout formats: InDesign ICML
- Outline formats: OPML
- TeX formats: LaTeX, ConTeXt, LaTeX Beamer slides
- PDF via LaTeX
- Lightweight markup formats: Markdown (including CommonMark), reStructuredText, AsciiDoc, MediaWiki markup, DokuWiki markup, Emacs Org-Mode, Textile
- Custom formats: written in lua.

RMARKDOWN (+ PANDOC)

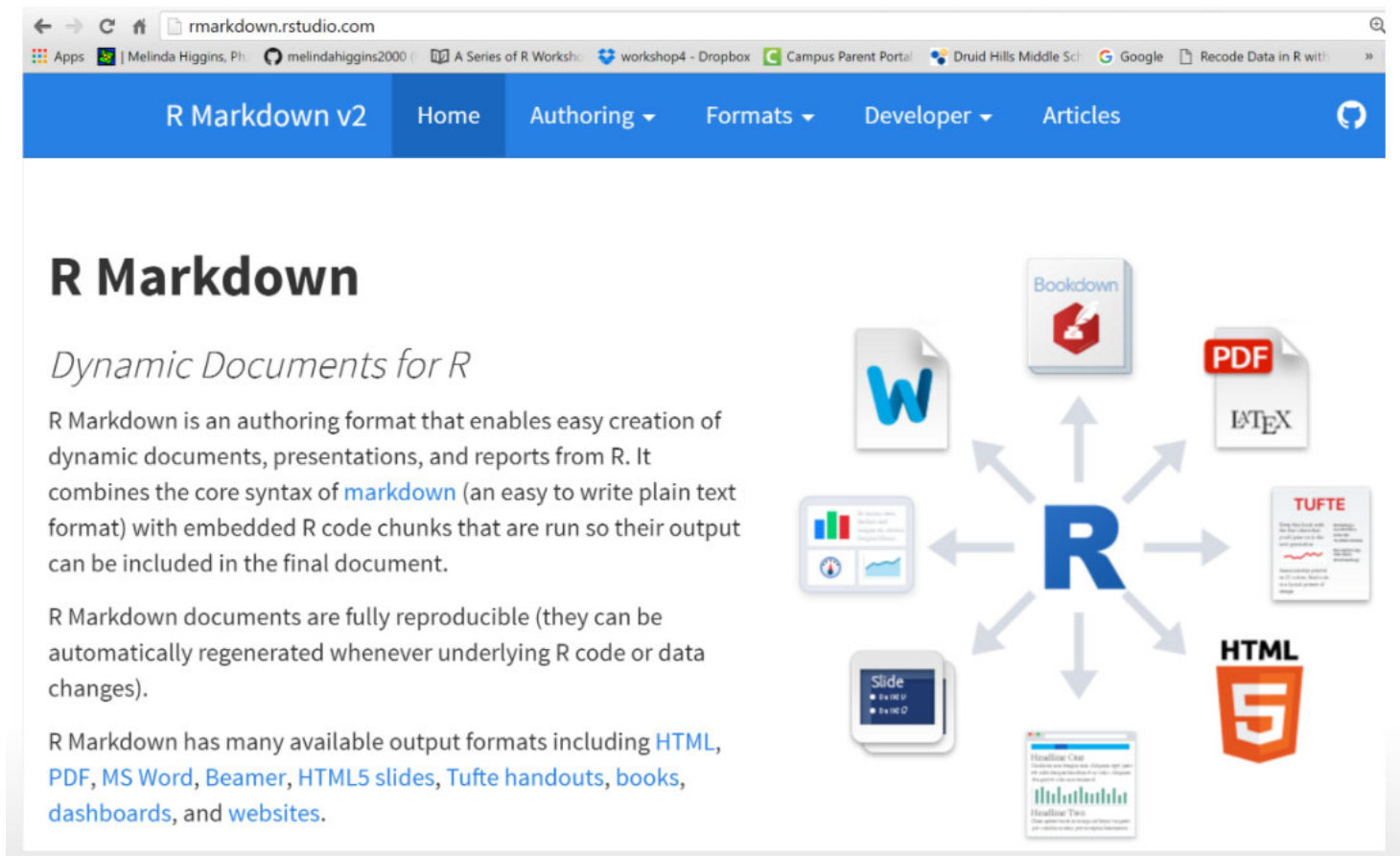
How it works



<https://rmarkdown.rstudio.com/>



R MARKDOWN “HUB”



The screenshot shows the R Markdown v2 website interface. The browser address bar displays `rmarkdown.rstudio.com`. The navigation bar includes links for Home, Authoring, Formats, Developer, and Articles. The main content area features the heading "R Markdown" and the subtitle "Dynamic Documents for R". The text describes R Markdown as an authoring format for creating dynamic documents, presentations, and reports, combining markdown syntax with embedded R code chunks. It lists various output formats: HTML, PDF, MS Word, Beamer, HTML5 slides, Tufte handouts, books, dashboards, and websites. A diagram on the right illustrates the R Markdown workflow, with a central "R" icon connected to icons for Word, Bookdown, PDF, Tufte, HTML, a dashboard, a slide, and a Beamer presentation.

R Markdown v2 Home Authoring Formats Developer Articles


R Markdown

Dynamic Documents for R

R Markdown is an authoring format that enables easy creation of dynamic documents, presentations, and reports from R. It combines the core syntax of [markdown](#) (an easy to write plain text format) with embedded R code chunks that are run so their output can be included in the final document.

R Markdown documents are fully reproducible (they can be automatically regenerated whenever underlying R code or data changes).

R Markdown has many available output formats including [HTML](#), [PDF](#), [MS Word](#), [Beamer](#), [HTML5 slides](#), [Tufte handouts](#), [books](#), [dashboards](#), and [websites](#).



THE RSTUDIO IDE

The screenshot displays the RStudio IDE interface with the following components:

- Source Panel (Top Left):** Shows an R Markdown file named `lesson3.Rmd`. The text includes a title "Getting started Git, Github and Data Wrangling" and instructions about installing Git and creating a Github account.
- Environment Panel (Top Right):** Displays a list of files in the current project, including `Contrastive.html`, `Contrastive_files`, `README.html`, `UCI.html`, `article.html`, `articleClique.html`, `bagboost.html`, `break.html`, `clusterclass.html`, and `colophon.html`.
- Console Panel (Bottom Left):** Shows the R command prompt output, including the R license text and instructions on how to use R, such as `license()`, `licence()`, `contributors()`, `citation()`, `demo()`, `help()`, `help.start()`, and `q()`.
- Help Panel (Bottom Right):** Displays the R documentation for the `factor` class, including the description and usage instructions.

R Scripts, RMD, ...

Environment History, GIT, ...

CONSOLE

Files, Plots, Packages, Help, Other Output

RMARKDOWN

YAML

```
1 ---
2 title: "Abalones"
3 author: "Melinda Higgins"
4 date: "November 10, 2019"
5 output: html_document
6 ---
```

R CODE
CHUNK

```
8 ```{r setup, include=FALSE}
9 # set up knitr options for all code chunks
10 knitr::opts_chunk$set(echo = FALSE)
11 knitr::opts_chunk$set(message = FALSE)
12 knitr::opts_chunk$set(warning = FALSE)
13 knitr::opts_chunk$set(error = FALSE)
14
15 # load packages needed for code chunks
16 library(readr)
17 library(tibble)
18 library(dplyr)
19 library(ggplot2)
20
21 # import abalone.csv dataset
22 # use read_csv()
23 # function from readr to import data
24 abalone <- read_csv("abalone.csv")
25 ```
```

“Marked” up text

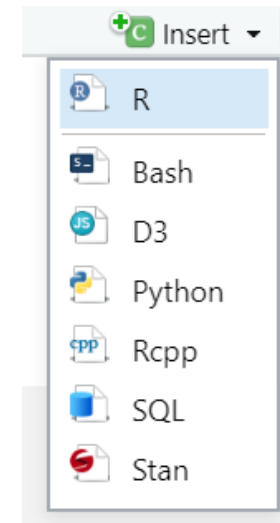
```
27 ## A Glimpse of the Abalone Dataset
```

```
28
29 Use the `glimpse()` function from the `tibble` package to take a peak
30 at the abalone dataset.
```



DEMOS

NOT JUST FOR R ANYMORE...



```
> library(bookdown)
> names(knitr::knit_engines$get())
```

[1] "awk"	"bash"	"coffee"	"gawk"	"groovy"
[6] "haskell"	"lein"	"mysql"	"node"	"octave"
[11] "perl"	"psql"	"Rscript"	"ruby"	"sas"
[16] "scala"	"sed"	"sh"	"stata"	"zsh"
[21] "highlight"	"Rcpp"	"tikz"	"dot"	"c"
[26] "cc"	"fortran"	"fortran95"	"asy"	"cat"
[31] "asis"	"stan"	"block"	"block2"	"js"
[36] "css"	"sql"	"go"	"python"	"julia"
[41] "sass"	"scss"	"theorem"	"lemma"	"corollary"
[46] "proposition"	"conjecture"	"definition"	"example"	"exercise"
[51] "proof"	"remark"	"solution"		

R Interface to Python

The **reticulate** package provides a comprehensive set of tools for interoperability between Python and R. The package includes facilities for:

- Calling Python from R in a variety of ways including R Markdown, sourcing Python scripts, importing Python modules, and using Python interactively within an R session.
- Translation between R and Python objects (for example, between R and Pandas data frames, or between R matrices and NumPy arrays).
- Flexible binding to different versions of Python including virtual environments and Conda environments.



Reticulate embeds a Python session within your R session, enabling seamless, high-performance interoperability. If you are an R developer that uses Python for some of your work or a member of data science team that uses both languages, reticulate can dramatically streamline your workflow!

MORE THAN R AND PYTHON...



NOVEMBER
2020

15 Other Languages

- 15.1 Register a custom language ...
- 15.2 Run Python code and interac...
- 15.3 Execute content conditionall...
- 15.4 Execute Shell scripts
- 15.5 Visualization with D3
- 15.6 Write the chunk content to a ...
- 15.7 Run SAS code
- 15.8 Run Stata code
- 15.9 Create graphics with Asympt...
- 15.10 Style HTML pages with Sas...

<https://bookdown.org/yihui/rmarkdown-cookbook/other-languages.html>

MORE THAN R AND PYTHON...

15.7 Run SAS code

You may run SAS (<https://www.sas.com>) code using the `sas` engine. You need to either make sure the SAS executable is in your environment variable `PATH`, or (if you do not know what `PATH` means) provide the full path to the SAS executable via the chunk option `engine.path`, e.g., `engine.path = "C:\\Program Files\\SASHome\\x86\\SASFoundation\\9.3\\sas.exe"`. Below is an example to print out "Hello World":

```
```${sas}  
data _null_;
put 'Hello, world!';
run;
```
```

Also see

<https://www.ssc.wisc.edu/~hemken/SASworkshops/Markdown/SASmarkdown.html>
<https://cran.r-project.org/web/packages/SASmarkdown/>

MORE THAN R AND PYTHON...

15.8 Run Stata code

You can run Stata (<https://www.stata.com>) code with the `stata` engine if you have installed Stata. Unless the `stata` executable can be found via the environment variable `PATH`, you need to specify the full path to the executable via the chunk option `engine.path`, e.g., `engine.path = "C:/Program Files (x86)/Stata15/StataSE-64.exe"`. The following is a quick example:

```
``{stata}
sysuse auto
summarize
``
```

The `stata` engine in knitr is quite limited. Doug Hemken has substantially extended it in the Statamarkdown package, which is available on GitHub at <https://github.com/Hemken/Statamarkdown>. You may find tutorials about this package by searching online for “Stata R Markdown.”

QUESTIONS?

My contact info:

Melinda.higgins@emory.edu

<https://melindahiggins.netlify.app/>