

Code-centric Websites with Blogdown & R

Best of both Worlds

Ilja / @fubits

Berlin R Users Group MeetUp, 2018/11/29

Outline

1. Why Blog?
2. (R) Markdown & Blogdown
3. Demo / Hands-On
 - Setup
 - Write / Code / Publish
 - Customize
4. Why Hugo/Static?

Why? Portfolio!

Why? Portfolio!

"if you don't have a website nowadays, you don't exist" (Scheidegger 2012 via Xie 2018)

*"While a resume matters, having a portfolio of **public evidence of your data science skills** can do wonders for your job prospects. Even if you have a referral, the ability to show potential employers what you can do instead of just telling them you can do something is important." (Galarnyk 2018)*

Why? Portfolio!

"if you don't have a website nowadays, you don't exist" (Scheidegger 2012 via Xie 2018)

*"While a resume matters, having a portfolio of **public evidence of your data science skills** can do wonders for your job prospects. Even if you have a referral, the ability to show potential employers what you can do instead of just telling them you can do something is important." (Galarnyk 2018)*



(Galarnyk 2018: How to Build a Data Science Portfolio)

Why? Visibility!

"The most effective strategy for me was doing public work. I blogged and did a lot of open source development late in my PhD, and these helped give public evidence of my data science skills." (@drob Robinson 2018)

Why? Visibility!

*"The most effective strategy for me was doing **public work**. I blogged and did a lot of open source development late in my PhD, and these helped give public evidence of my data science skills." (@drob Robinson 2018)*

*"A well-designed and maintained website can be extremely helpful for other people to **know you**, and you do not need to wait for suitable chances at conferences or other occasions to introduce yourself in person to other people." (Xie 2018)*

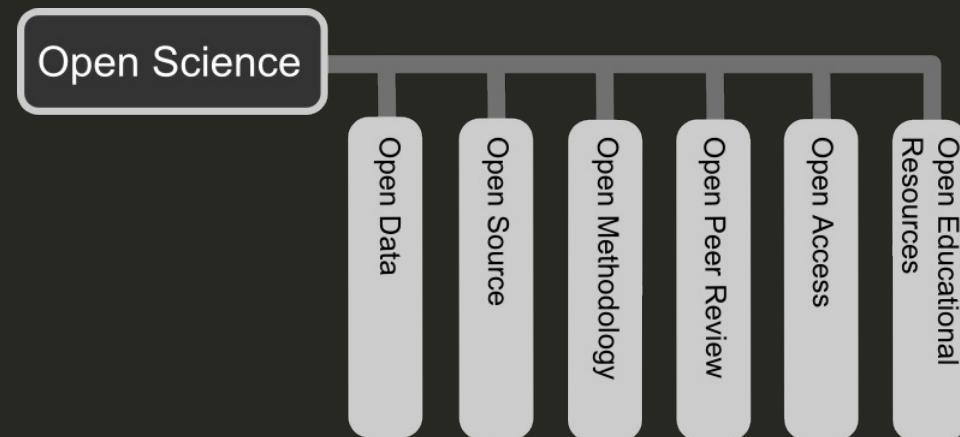
*"On the other hand, a website is also highly useful for yourself to **keep track** of what you have done and thought. Sometimes you may go back to a certain old post of yours to **relearn the tricks or methods** you once mastered in the past but have forgotten." (Xie 2018)*

Why? Open Source, Open Science, Replication!

Sharing is caring!

Why? Open Source, Open Science, Replication!

Sharing is caring!



(Source: Andreas E. Neuhold, CC BY 3.0, [Link](#))

The `selmpl(path.expand(input), verbose, fromFile)` issue has been fixed #1

 **Closed** yihui opened this issue on 7 Jul · 1 comment



yihui commented on 7 Jul

+ ...

FYI the `blogdown::serve_site()` problem you mentioned in your excellent blog post
<https://ellocke.github.io/post/r-troubleshooting-blogdown-hugo-for-windows-dummies/> should be fixed in
blogdown v0.7 (which I just published to CRAN): <https://github.com/rstudio/blogdown/releases/tag/v0.7>
Thanks!



ellocke commented on 28 Sep

Owner + ...

Thanks for the shout-out! I finally edited the post.



ellocke closed this on 28 Sep



ellocke added a commit that referenced this issue on 28 Sep

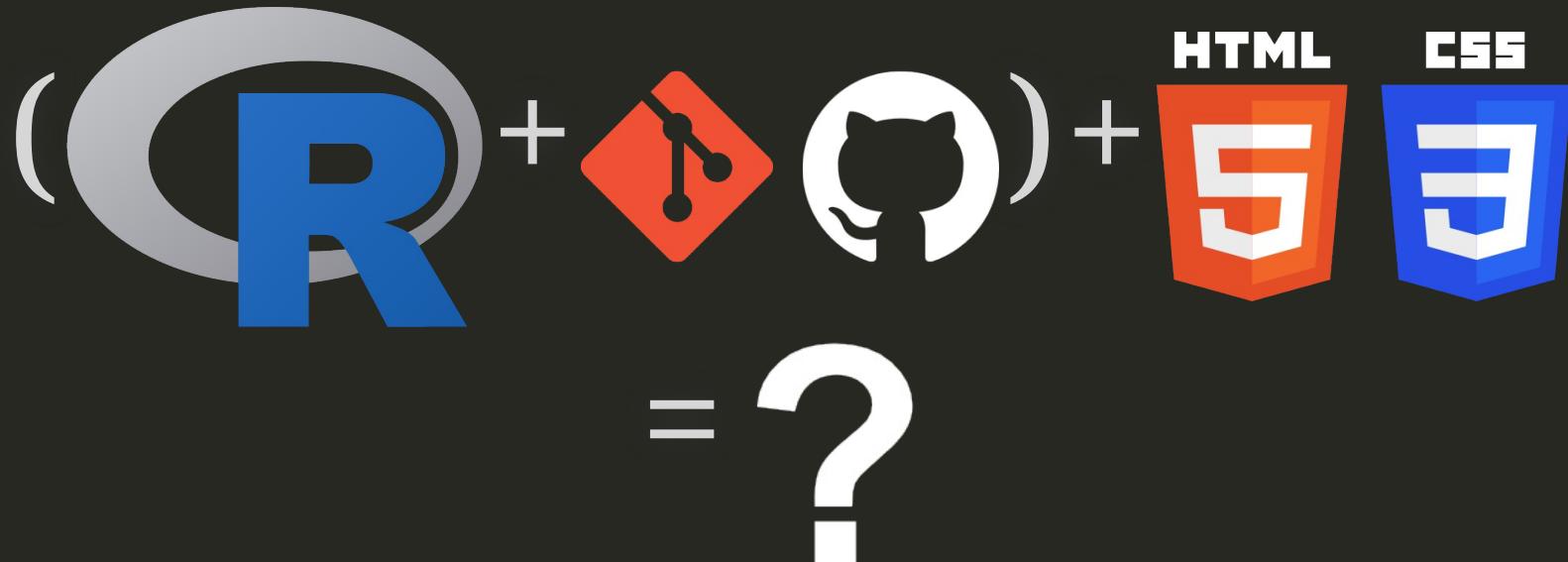


ellocke resolved Issue #1

6fdb55a

The Formula

The Formula



First, there was Markdown

First, there was Markdown

Headings (H1-H6)

First, there was Markdown

Headings (H1-H6)

plain text, **bold**, *italic*, ~~strike-through~~, [URL](<https://example.com>)

First, there was Markdown

Headings (H1-H6)

plain text, **bold**, *italic*, ~~strike-through~~, [URL](<https://example.com>)

| > Blockquotes

First, there was Markdown

Headings (H1-H6)

plain text, **bold**, *italic*, ~~strike-through~~, [URL](<https://example.com>)

| > Blockquotes

- 1. lists
 - o + nested lists

First, there was Markdown

Headings (H1-H6)

plain text, **bold**, *italic*, ~~strike-through~~, [URL](<https://example.com>)

| > Blockquotes

1. lists

o + nested lists

pretty `inline code` chunks

First, there was Markdown

Headings (H1-H6)

plain text, **bold**, *italic*, ~~strike-through~~, [URL](<https://example.com>)

| > Blockquotes

1. lists

o + nested lists

pretty `inline code` chunks

```
block code chunks
fun()
```

First, there was Markdown

Headings (H1-H6)

plain text, **bold**, *italic*, ~~strike-through~~, [URL](<https://example.com>)

| > Blockquotes

1. lists

o + nested lists

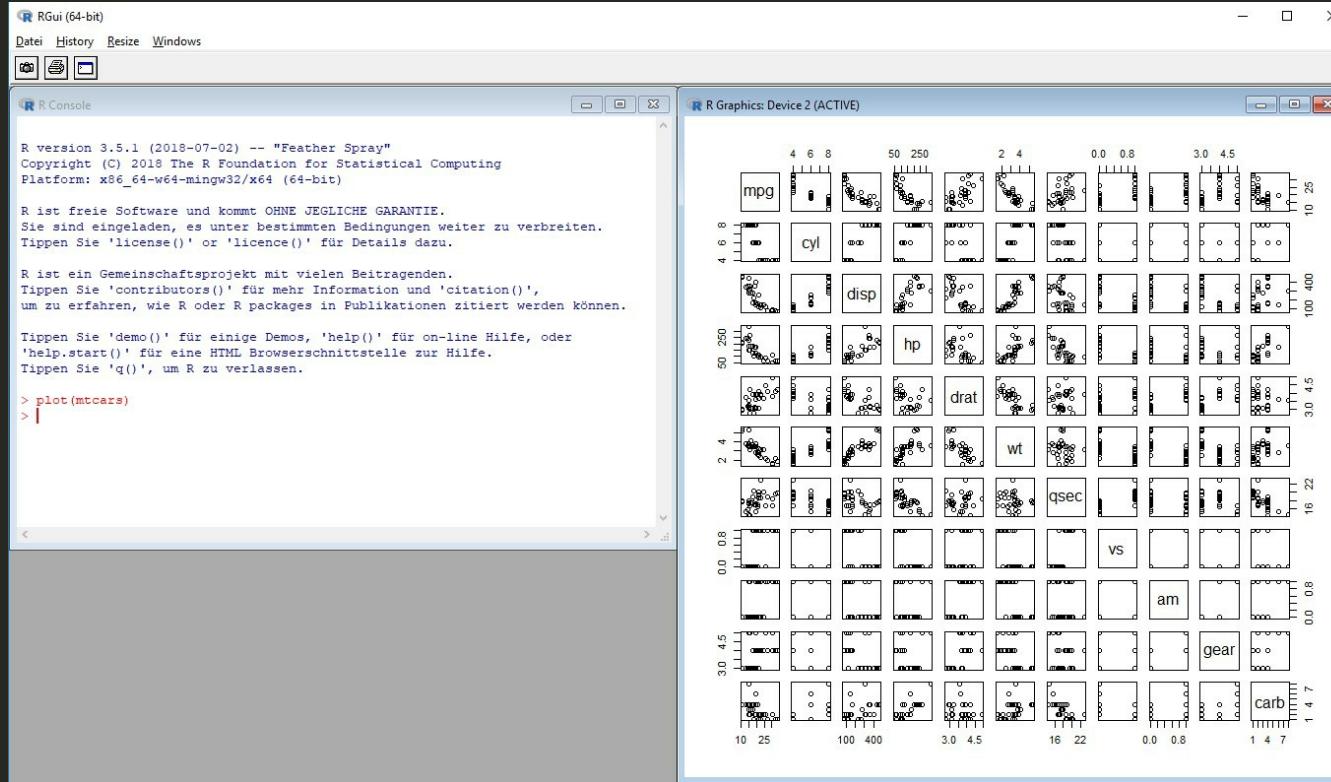
pretty `inline code` chunks

```
block code chunks
fun()
```

LaTex-like formulas:

$$\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i$$

Then there was Base R / R GUI



And then came RStudio Yihui Xie

And then came RStudio Yihui Xie



And then came RStudio Yihui Xie



- software engineer at RStudio (since 2013)
- PhD from the Dept. of Statistics, Iowa State University
- Twitter: [@xieyihui](https://twitter.com/xieyihui)
- Initiator of Chinese R Conference (in 2008!)



"YihuiVerse" R Rmarkdown Ecosystem

yihui.name/en/

github.com/yihui/



R + Markdown = RMarkdown

R + Markdown = RMarkdown

*"With R Markdown, you only need to maintain the **source** documents; all output pages can be automatically generated from source documents. This makes it much easier to maintain a website, especially when the website is related to **data analysis or statistical computing and graphics**.*

*When the source code is updated (e.g., the model or data is changed), your web pages can be **updated** accordingly and **automatically**. There is no need to run the code separately and cut-and-paste again. Besides the convenience, you gain **reproducibility** at the same time." (Xie 2018)*

R + Markdown = RMarkdown

*"With R Markdown, you only need to maintain the **source** documents; all output pages can be automatically generated from source documents. This makes it much easier to maintain a website, especially when the website is related to **data analysis or statistical computing and graphics**.*

*When the source code is updated (e.g., the model or data is changed), your web pages can be **updated** accordingly and **automatically**. There is no need to run the code separately and cut-and-paste again. Besides the convenience, you gain **reproducibility** at the same time." (Xie 2018)*

- Markdown+ Text
- **Code/Syntax** + Output
- shareable **.Rmd**
- Code + Layout are completely separated
- One Workflow / a single source / a single IDE
- Portability
- Version control

R Markdown : : CHEAT SHEET

What is R Markdown?

.Rmd File - An R Markdown (.Rmd) file is a record of your research. It contains the code that a scientist uses to produce their work along with the narration that a reader needs to understand your work.

Reproducible Research - At the click of a button, a reader of your work can rerun the code in an R Markdown file to reproduce your work and export the results.

Dynamic Documents - You can choose to export the finished report in a variety of formats, including html, pdf, MS Word, or RTF documents; html or pdf based slides; Notebooks; and more.

Workflow

- Open a new .Rmd file at File > New File > R Markdown. Use the wizard that opens to pre-populate the file with a template.
- Write document by editing template
- Knit document to create report; use knit button (red dot) to knit.
- Preview Output in IDE window
- Publish (optional) to web server
- Examine build log in R Markdown console
- Use output file that is saved along side .Rmd

Embed code with knitr syntax

INLINE CODE
Insert with `<code>`. Results appear as text without code.
Built with `r gtVersion()`. Built with 3.2.3

CODE CHUNKS
One or more lines surrounded with `{{`{r}}`}` and `}`. Place chunk options within curly braces, after r. Insert with

GLOBAL OPTIONS
Set with `knitr::opts_chunk\$set()`, e.g.
`{{`{r include=FALSE}}`}`

IMPORTANT CHUNK OPTIONS

cache - cache results for future knits (default = FALSE)	dependson - chunk dependencies for caching (default = NULL)	fig_align - left, right, or center (default = default)	message - display code messages in document (default = TRUE)
cache.path - directory to save cached results in (default = "cache")	echo - display code in output document (default = TRUE)	fig.cap - figure caption as character string (default = NULL)	results (default = "markup")
child - file(s) to knit and then include (default = NULL)	engine - code language used in chunk (default = "R")	fig.height, fig.width - Dimensions of plots in inches	hide - do not display results
collapse - collapse all output into single block (default = FALSE)	error - display error messages in doc (TRUE) or suppress when errors occur (FALSE) (default = FALSE)	highlight - highlight source code (default = TRUE)	keep - put all results below all code
comment - prefix for each line of results (default = "#")	eval - Run code in chunk (default = TRUE)	include - include chunk in doc after running (default = TRUE)	tidy - tidy up for display (default = FALSE)
		warning - display code warnings in document (default = TRUE)	warning - display code warnings in document (default = TRUE)

Options not listed above: `r.options`, `arrows`, `autodev`, `background`, `cache.comments`, `cache.lazy`, `cache.rebuild`, `cache.vars`, `dev`, `devargs`, `dpi`, `engine.opts`, `engine.opts.chunk`, `engine.opts.out`, `engine.opts.out.extra`, `engine.opts.out.height`, `engine.opts.out.width`, `engine.opts.prompt`, `engine.opts.reflabel`, `engine.opts.render`, `engine.opts.size`, `split`, `tidy.opts`

render

Use `markdown::render()` to render/knit at cmd line. Important args:
input - file to render
output_options - List of render options (as in YAML)
output_file - output file
output_dir - output directory

params - list of params to use
envir - environment to evaluate code chunks in
encoding - input file

R Markdown

This is an R Markdown document. Markdown is a simple syntax for authoring HTML, PDF, and MS Word documents.

```
## speed
## Min. : 1.4.0
## 1st Qu.:12.0
## Median :15.0
## 3rd Qu.:24.0
## Max. :120.0
```

For more details on using R Markdown see <http://rmarkdown.rstudio.com/>.

R Markdown Structure

.rmd Structure

YAML Header
Options for the render (e.g. pandoc) options written as key/value pairs (YAML).

At start of file
Between lines of ...

Text
Narration formatted with markdown, mixed with:

Code Chunks
Chunks of embedded code. Each chunk:
Begins with `{{`{r}}`}`
ends with `}`

R Markdown will run the code and append the results to the doc. It will use the location of the .Rmd file as the working directory.

Parameters

Parameterize your documents to reuse with different inputs (e.g., data, values, etc.)

- Add parameters** - Create and set parameters in the header as sub-values of `params`
- Call parameters** - Call parameter values in code as `params$name`
- Set parameters** - Set values with `knitWithParams` or the `params` argument of `render`:

```
render("doc.Rmd", params = list(a = 1,
d = Sys.Date()))
```

Today's date is "params\$d"

Interactive Documents

Turn your report into an interactive Shiny document in 4 steps

- Add runtime: shiny to the YAML header.
- Call Shiny input functions to embed input objects.
- Call Shiny render functions to embed reactive output.
- Render with `rmarkdown::run` or click Run Document in RStudio IDE

Shiny

Embed a complete app into your document with `shiny::shinyAppDir()`

NOTE: Your report will rendered as a Shiny app, which means you must choose an HTML output format, like `html_document`, and serve it with an active R Session.

How many cars?

	speed	dist
1	4.00	2.00
2	4.00	10.00
3	7.00	4.00
4	7.00	22.00
5	8.00	16.00

RStudio: R Markdown Cheat Sheet

13 / 25

Documentation



- <https://bookdown.org/yihui/rmarkdown/>
- <https://bookdown.org/yihui/bookdown/>
- <https://bookdown.org/yihui/blogdown/>

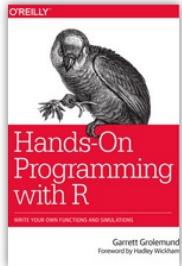
Bookdown

Hands-On Programming with R

by Garrett Grolemund

2018-11-21*

Star 75



This book will teach you how to program in R, with hands-on examples. I wrote it for non-programmers to provide a friendly introduction to the R language. You'll learn how to load data, assemble and disassemble data objects, navigate R's environment system, write your own functions, and use all of R's programming tools. Throughout the book, you'll use your newfound skills to solve practical data science problems. *Read more →*

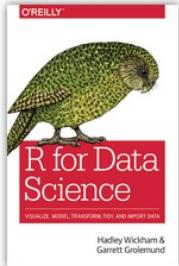
15

R for Data Science

by Garrett Grolemund, Hadley Wickham

2018-11-21*

Star 1,629



This book will teach you how to do data science with R: You'll learn how to get your data into R, get it into the most useful structure, transform it, visualise it and model it. In this book, you will find a practicum of skills for data science. Just as a chemist learns how to clean test tubes and stock a lab, you'll learn how to clean data and draw plots—and many other things besides. These are the skills that allow data science to happen, and here you will find the best practices for doing each of these things with R. You'll learn how to use the grammar of graphics, literate programming, and reproducible research to save time. You'll also learn how to manage cognitive resources to facilitate discoveries when wrangling, visualising, and exploring data. *Read more →*

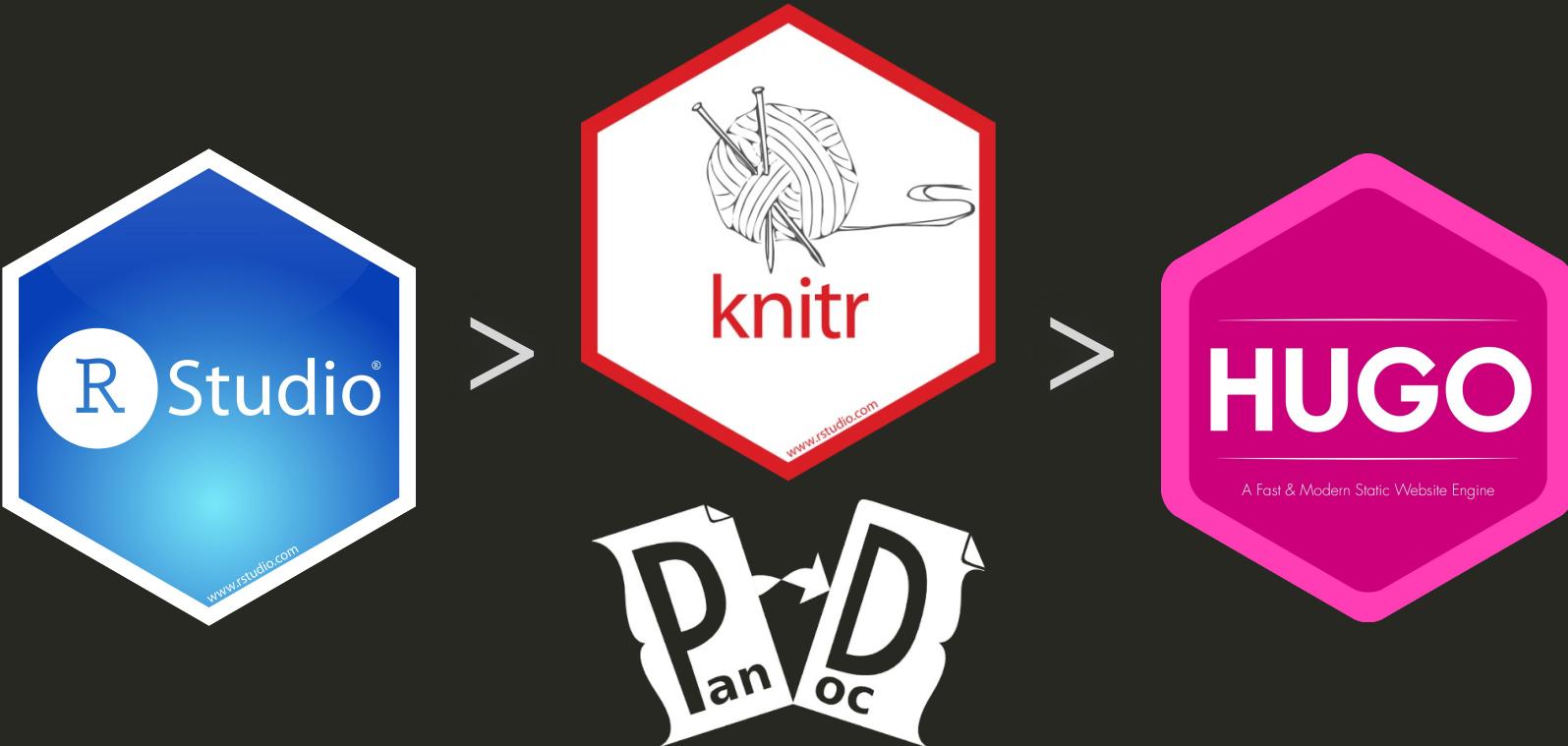
16

<https://r4ds.had.co.nz/>

15 / 25

The Formula!

The Formula!



= Blogdown



www.rstudio.com

Blogdown: Setup

Blogdown: Setup

I. Install Blogdown

```
install.packages("blogdown")
```

Blogdown: Setup

I. Install Blogdown

```
install.packages("blogdown")
```

II. Install Hugo

```
blogdown::install_hugo()
```

Blogdown: Setup

I. Install Blogdown

```
install.packages("blogdown")
```

II. Install Hugo

```
blogdown::install_hugo()
```

III. RStudio: New Site

```
File -> New Project -> Website using Blogdown
```

OR

```
blogdown::new_site()
```

Blogdown: Setup

I. Install Blogdown

```
install.packages("blogdown")
```

II. Install Hugo

```
blogdown::install_hugo()
```

III. RStudio: New Site

```
File -> New Project -> Website using Blogdown
```

OR

```
blogdown::new_site()
```

- There are tons of excellent how-to's, i.e. by [Tyler Clavelle](#)

2+1 Commands To Rule Them All

2+1 Commands To Rule Them All

I. Preview locally with/out visibility in local network

```
blogdown::serve_site(host = '0.0.0.0') / blogdown::serve_site()
```

2+1 Commands To Rule Them All

I. Preview locally with/out visibility in local network

```
blogdown::serve_site(host = '0.0.0.0') / blogdown::serve_site()
```

II. Build to /public

```
blogdown::hugo_build()
```

2+1 Commands To Rule Them All

I. Preview locally with/out visibility in local network

```
blogdown::serve_site(host = '0.0.0.0') / blogdown::serve_site()
```

II. Build to /public

```
blogdown::hugo_build()
```

III. (Publish)

```
git commit -m "updated Twitter Post" / git push
```

Demo

Demo

I. Markdown Post

Demo

I. Markdown Post

II. Blogdown Post

Demo

I. Markdown Post

II. Blogdown Post

III. Theme + CSS

Demo

I. Markdown Post

II. Blogdown Post

III. Theme + CSS

*"If you do not understand HTML, CSS, or JavaScript, and have no experience with Hugo themes or templates, it may take you about **10 minutes to get started** with your new website, since you have to accept everything you are given (such as the default theme); if you do have the knowledge and experience (and desire to highly customize your site), it may take you **several days** to get started. Hugo is really powerful. Be cautious with power." (Xie et al. 2018, Ch. 1.6)*

Demo

I. Markdown Post

II. Blogdown Post

III. Theme + CSS

*"If you do not understand HTML, CSS, or JavaScript, and have no experience with Hugo themes or templates, it may take you about **10 minutes to get started** with your new website, since you have to accept everything you are given (such as the default theme); if you do have the knowledge and experience (and desire to highly customize your site), it may take you **several days** to get started. Hugo is really powerful. Be cautious with power."* (Xie et al. 2018, Ch. 1.6)



(cf. modded theme with [source](#))

More Features

- Shortcodes (i.e. `{{< tweet 989470885475008512 >}}`)
- CSS + JS
- RSS
- Source File
- `static/other`
 - i.e. `slides` for this talk
 - `R/build.R` from `/static/talks/index.Rmd`

- **Widgets**

Tips & Tweaks

- set up Hugo properly -> `hugo -v`
- change `/public` folder (cf. **Blogdown Book**)
- use `here::here("static","data")`
- Windows: Dual boot Linux OR put the Project Folder in root (255 path length limit!)
- use Git
- save objects with dynamic data (i.e. from API) locally
- DO NOT SAVE LARGE OBJECTS TO `/static/data`

Why Static?

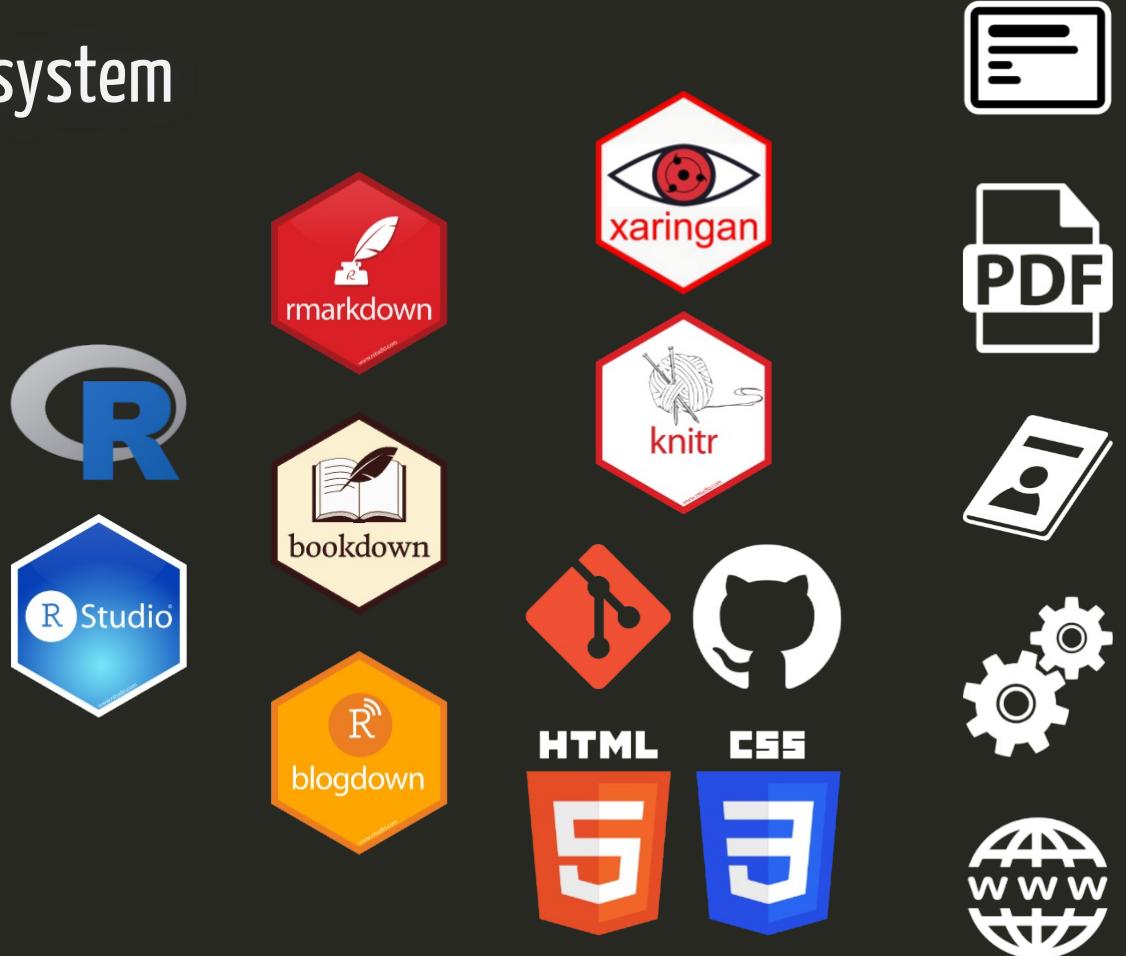
- Benefits of Static Site Generators
- Ownership + Portability
- no DB & no Services (PHP et al.)
- no CMS + local building & testing
- Security (+2FA GitHub)
- GDPR compliance (i.e. DB, external CDN) (see [Hugo-specific discussion & implementation](#))

The R Rmarkdown Ecosystem



The R Rmarkdown Ecosystem

- A single Workflow
- A single IDE
- A Single Source
- {.Rmd, Notebook, Slides, PDF, Book}
- Version Control
- Open Science!



Thanks!

dadascience.design/talks

@fubits

CC BY-SA 4.0

Slides created via the R package **xaringan**.