

Code Club: R Markdown (from Rstudio)

Martin Schobben

January 23, 2019



Why R Markdown?

- ▶ Merge data, code and text
 - ▶ resulting in less errors (instead of `Ctrl+C`–`Ctrl+V`)
- ▶ Good reproducibility
- ▶ Formatting is performed only at the end
 - ▶ creating templates (reports, manuscripts, presentations ...)
- ▶ Easy collaboration
 - ▶ output: HTML, PDF, MS word and lot's more
 - ▶ instead of only PDF in \LaTeX



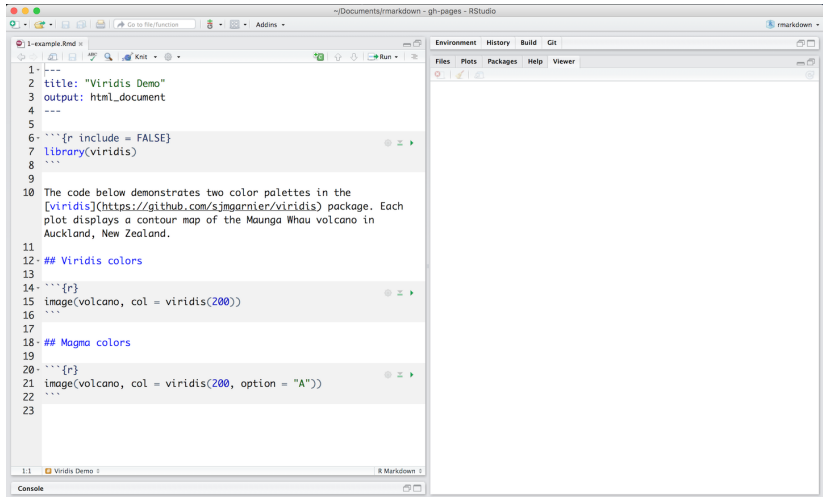
How it works



- ▶ The R package knitr forms the backbone for knitting in Rstudio
- ▶ Pandoc processes the markdown file
- ▶ The creator of Rmarkdown from Rstudio and knitr: Yihui Xie (software engineer at RStudio)



R Markdown from Rstudio <http://rmarkdown.rstudio.com>



```
1- example.Rmd x
1 ---
2 title: "Viridis Demo"
3 output: html_document
4 ---
5
6 ```{r include = FALSE}
7 library(viridis)
8 ```
9
10 The code below demonstrates two color palettes in the
11 [viridis](https://github.com/sjmgarnier/viridis) package. Each
12 plot displays a contour map of the Maunga Whau volcano in
13 Auckland, New Zealand.
14
15 ## Viridis colors
16
17 ```{r}
18 image(volcano, col = viridis(200))
19 ```
20
21 ## Magma colors
22
23 ```{r}
24 image(volcano, col = viridis(200, option = "A"))
25 ```
26
27 1:1 Viridis Demo | R Markdown
28 Console
```

R Markdown from Rstudio <http://rmarkdown.rstudio.com>

R Markdown

from RStudio

Get Started Gallery Formats Articles Book

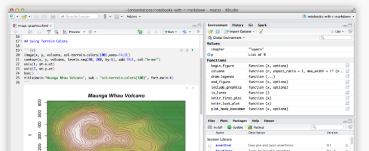


Analyze. Share. Reproduce.

Your data tells a story. Tell it with R Markdown.

Turn your analyses into high quality documents, reports, presentations and dashboards.

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.



Utrecht University

R Markdown from Rstudio <http://rmarkdown.rstudio.com>

R Markdown Cheat Sheet

learn more at rmarkdown.rstudio.com



.Rmd files

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



Reproducible Research

At the click of a button, or the type of a command, you can run the code in an R Markdown file to reproduce your work and export the results as a finished report.



Dynamic Documents

You can choose to export the finished report as a html, pdf, MS Word, ODT, RTF, or markdown document; or as a html or pdf based slide show.

Workflow

- 1 Open a new .Rmd file at File > New File > R Markdown. Use the wizard that opens to pre-populate the file with a template.
- 2 Write document by editing template.
- 3 Knit document to create report. Use knit button (or render!) to knit.
- 4 Preview Output in IDE window.
- 5 Publish (optional) to web or server. Publish button to accounts at:
 - GitHub.com
 - RStudio ConnectReload document.
- 6 Examine build log in R Markdown console.
- 7 Use output file that is saved alongside .Rmd.

.Rmd structure

YAML Header
Optional section of header (e.g. pandoc) options written as keyvalue 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

Workflow
Open in window, Save, Spell Check, Find and replace, Publish, Show outline, Insert chunk, Go to code chunk, Run code chunk(s), Modify chunk options, Run all previous chunks, Run current chunk.

Knit document to create report
Use knit button (or render!) to knit.

Preview Output in IDE window

Publish (optional)
Publish button to accounts at:

- GitHub.com
- RStudio Connect

Reload document.

Examine build log in R Markdown console

Use output file that is saved alongside .Rmd

render()
Use markdown::render() to render/knit at cmd line.
Important args:
input: file to render
output_format: output options - List of render options (as in YAML)
output_dir: output_dir
params: list of params to use
envir: environment to evaluate code chunks in
encoding: of input file

Interactive Documents

Turn your report into an interactive Shiny document in 4 steps

- 1 Add runtime: shiny to the YAML header.
- 2 Call shiny input functions to embed input objects.
- 3 Call shiny render functions to embed reactive output.
- 4 Render with markdown::run or click Run Document in RStudio IDE

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

* 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.

Embed code with knitr syntax

Code chunks
One or more lines surrounded with ````{r}` and `````. Place chunk options within curly braces, after ````{r}` insert with ````{r, options}`

Global options
Set with knitr::opts_chunk\$set(), e.g.
`knitr::opts_chunk$set(echo = TRUE)`

Inline code
Insert with `<r code>`. Results appear as text without code.
Built with `r<getVersion>()` → Built with 3.2.3

Important chunk options

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

Parameters

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

- 1 Add parameters
Create and set parameters in the header as sub-values of params
- 2 Call parameters
Call parameter values in code as `params$name`
- 3 Set parameters
Set values with `knitr::with_params()` or the params argument of `render()`:
`render("doc.Rmd", params = list(n = 1, d = as.Date("2015-01-01")))`

Today's date is `params$d`

Knit to HTML, Knit to PDF, Knit to Word, Knit with Parameters...

versity

This is an R Markdown presentation!!

