2017-01-30

Rick O. Gilmore

2017-01-29 09:13:19

Objectives

- Learn how to render multiple output formats from a single R Markdown file

Use cases

- You want to give a talk based on your analysis
- You want to give a collaborator a report on your analysis in a format they prefer
- You don't want to duplicate effort

Output options

- Document
  - HTML
    - Embedded or non-embedded figures
  - MS Word
  - <PDF

Output options

- Presentation
  - HTML (ioslides)
  - HTML (Slidy)
  - PDF (Beamer)

Output options

- Shiny (interactive document)
  - Document
  - Presentation (ioslides)

Output options

- A website!

All in your head(er)

- RMarkdown documents have header text written in YAML
- YAML = YAML Ain't Markup Language

```
---  title: "One Code to Rule Them All"  author: "Rick O. Gilmore"  date: "July 4, 1776"
output: html_document ---
```

Default output is `html_document`

- `rmarkdown::render(input = "my-report.Rmd")` creates a single HTML file
- The `output: html_document` line specifies the default format for the file
- Omitting the `output:` or `author:` or `date:` commands still yields an HTML file

Getting fancier

- `rmarkdown::render()` can take an `output_format` parameter
- `rmarkdown::render(input = "my-report.Rmd", output_format = "html_document")` renders an HTML file as before
- `rmarkdown::render(input = "my-report.Rmd", output_format = "word_document")` renders an MS Word document
- `rmarkdown::render(input = "my-report.Rmd", output_format = "pdf_document")` renders a PDF document

```
rmarkdown::render(input = "my-report.Rmd", output_format = "html_document") rmarkdown::render(input
= "my-report.Rmd", output_format = "word_document") rmarkdown::render(input = "my-report.Rmd",
output_format = "pdf_document")
```

Or, just

```
rmarkdown::render(input = "my-report.Rmd", output_format = c("html_document", "word_document",
"pdf_document")
```

makes all three with **one** command.

Y'all can customize your YAML

- Default `html_document` output creates a single HTML file with all figures embedded via
  - `self_contained: true`
- Nice for sharing a single file with collaborators
- Not so if you want to save the figures for other uses
- If `self_contained: false`, then figures are written to separate directory

```
---  title: "One Code to Rule Them All"  author: "Rick O. Gilmore"  date: "July 4, 1776"
output: html_document: self_contained: false ---
```

Exploring report parameters

Presentations

- **HTML (ioslides)**
- HTML (Slidy)
- PDF (Beamer)

ioslides basics

- New slides with titles start with double-hashes

```
## New Slide Title
```

- New slides without titles start with four dashes

```
----
```

ioslides basics

- Bullet points start with single dashes

```
- Bulleted item 1 - Bulleted item 2
```

- if `incremental: FALSE` in YAML header (the default), then

```
>- Incremental bullet
```

allows you to show one bullet at a time for that slide

ioslides basics

- Embedding figures requires some HTML

`<img src="fig.jpg">` embeds

- Embedding videos depends on source
  - YouTube provides code
  - Embedding other (even local) videos requires HTML

ioslides formatting

```
<div class="centered">
...
</div>
```

Centers items between the `<div>` and `<\div>` tags vertically.

Rendering ioslides

```
---  title: "One Code to Rule Them All"  author: "Rick O. Gilmore"  date: "July 4, 1776"
output: ioslides_presentation ---
```

in your file header, or

```
rmarkdown::render(input = "my-report.Rmd", output_format = "ioslides_presentation")
```

on the fly.

Parameters for ioslides

`incremental: false`

Turns off incremental reveal of bullets.

`widescreen: true`

Makes slides work well with 16:9 aspect ratio screens.

Documents

- HTML documents, `output_format="html_document"`
- **MS Word documents,**`output_format="word_document"`
- PDF documents, `output_format="pdf_document"`

MS Word document basics

See http://rmarkdown.rstudio.com/word_document_format.html

```
---
...
output:
  word_document:
    fig_width: 5
    fig_height: 5
    fig_caption: false
    df_print: kable
    reference_docx: mystyles.docx
...
---
```

Rendering MS Word documents

`rmarkdown::render(input = "my-report.Rmd", output_format = "word_document")`

if your document header *does not* have parameters for word_document, or

`rmarkdown::render(input = "my-report.Rmd")`

if your document header has the parameters and `word_document` is first in the list.

PDF documents

See http://rmarkdown.rstudio.com/pdf_document_format.html

- Put in 1st position in header, then `rmarkdown::render(input = "my-report.Rmd")`
- Or, `rmarkdown::render(input = "my-report.Rmd", output_format = "pdf_document")`
- Must install LaTeX for this to work

```
---
...
  pdf_document:
```

```
    toc: true
    toc_depth: 2
...
---
```

Some useful parameters for the `rmarkdown::render()` command

- `output_file = myslides.ioslides.html` or `output_file = myslides.slidy.html` to specify different output file targets.
- `output_dir = reports` or `output_dir = docx` to direct output to a specific directory.
- `params = list(name = "Joe", quest = "Find the grail", favorite.color = "blue")` to pass parameters that R can use via `params$name`

How I used parameter passing like this

https://github.com/gilmore-lab/gilmore-thomas-fesi-2015/blob/master/render_all.R

https://raw.githubusercontent.com/gilmore-lab/gilmore-thomas-fesi-2015/master/gilmore-thomas-fesi.Rmd

Making a website in R Markdown

- Components (in a separate folder):
  - `_site.yml` parameter file
  - `index.Rmd` and other `*.Rmd` files. (N.B., the * character is a wildcard meaning it can represent any number of characters)

`_site.yml`

```
name: "A Site"
navbar:
  title: "Site Title"
  left:
    - text: "Home"
      href: index.html
```

`index.Rmd`

```
---
title: "Website home"
---

My website's home page.
```

Note that the `index.Rmd` file looks like the simplest possible R Markdown file. There's nothing but the title in the header. Of course, you can put all kinds of text, images, videos, and such on each page.

Render this file into a web page with `rmarkdown:render_site()` or by pressing the `Knit` button.

My workflow for this course

- Make slides in R Markdown with `ioslides_presentation` as output option
- Modify `schedule.Rmd` file adding links to lecture notes and in-class activities:

Code snippet from `schedule.Rmd`

```
- **Lecture notes**. [HTML](lecture-notes/2017-01-30.html). [Rmd](lecture-notes/2017-01-30.Rmd). [PDF](
- **Activity notes**. [HTML](in-class-activities/2017-01-23.html). [Rmd](in-class-activities/2017-01-23
```

My workflow for this course

- Render slides into .html and .pdf versions
- Render site
  - Save site documents (.html +) into a special `docs/` folder
  - I wrote R functions to render slides and site

My workflow for this course

- Commit changes (save this version) and push to GitHub
- I've set up course repository (repo) so GitHub renders pages as a website with http://psu-psychology.github.io/psy-511-reproducible-research-spring-2017/ as the URL.
- Any repo can have a site like this

Your assignment

- Create a template for a reproducible research report **OR**
  - Rmd template
  - You might also convert your study protocol into an Rmd document.
- Create talk slides **OR**
  - Rmd template
- Create a draft project website
  - Rmd template

- Show us your work!
  - Bring computer to class and show locally **OR**
  - Send Rick a zip file archive **OR**
  - Email Rick your files **OR**
  - Print out your files **OR**
  - Push to a web site somewhere and send a link on our Slack channel **OR**
  - Add to your private GitHub repo for this course. https://github.com/psu-psych-511-2017-spring.