xaringan: Delivering Presentation with R Markdown

Emi Tanaka, Joseph Casillas, Eric Nantz, Yihui Xie

2019-03-09

Contents

Preface

Hello.

About the Authors

We are a team of shinobi and kunoichi who wish to share the fun and secrets of the **xaringan** package with you.

Emi Tanaka

Lead author, and the ninja theme author.

Joseph Casillas

Contributor of xaringan.

Eric Nantz

Host of the R Podcast.

Yihui Xie

Yihui Xie (https://yihui.name) is a software engineer at RStudio (https://www.rstudio.com). He earned his PhD from the Department of Statistics, Iowa State University. He is interested in interactive statistical graphics and statistical computing. As an active R user, he has authored several R packages, such as **knitr**, **bookdown**, **blogdown**, **xaringan**, **tinytex**, **animation**, **DT**, **tufte**, **formatR**, **fun**, **xfun**, **mime**, **highr**, **servr**, and **Rd2roxygen**, among which the **animation** package won the 2009 John M. Chambers Statistical Software Award (ASA). He also co-authored a few other R packages, including **shiny**, **rmarkdown**, and **leaflet**.

He has authored two books, *Dynamic Documents with knitr* (**?**), and *bookdown: Authoring Books and Technical Documents with R Markdown* (**?**), and co-authored the book, *blogdown: Creating Websites with R Markdown* (**?**).

In 2006, he founded the Capital of Statistics (https://cosx.org), which has grown into a large online community on statistics in China. He initiated the Chinese R conference in 2008, and has been involved in organizing R conferences in China since then. During his PhD training at Iowa State University, he won the Vince Sposito Statistical Computing Award (2011) and the Snedecor Award (2012) in the Department of Statistics.

He occasionally rants on Twitter (https://twitter.com/xieyihui, well not so much these days), and most of the time you can find him on GitHub (https://github.com/yihui).

He enjoys spicy food as much as classical Chinese literature and is a master hacker freeing the frustation of thousands of people that would have needed to repeatedly copy-and-paste the R output and saving science with the gift of reproducible research.

Chapter 1

Introduction

In late 2016, Yihui discovered remark.js (Bang 2018) and loved it at the first sight. A few weeks later in the R Markdown ecosystem (Allaire et al. 2018), an R package was born and named xaringan (???), which nobody knows how to pronounce (including Yihui himself, because it was adapted from the Japanese manga series Naruto by Kishimoto (2007)). Anyway, this package has gained some popularity, and some CSS ninja have started contributing themes to it. One day, Yihui was thinking about creating a gallery for existing themes in xaringan. After a few replies in the Github issue, he realized there might be enough topics on xaringan for a short book. Accidentally, he invented a new development model for writing books: the Github-issue-driven book development.

Ideas so far:

This could include the basics:

- Installation
- Motivation - Why slideshows using R Markdown (should mention about ioslides too and differences to it)
- Structure of the book
- etc

Chapter 2

Basics

In this chapter you will get up and moving with your first HTML presentation using `xaringan`. Specifically, we will cover the basics regarding getting started and generating your HTML slides, as well as talk about some markdown and `xaringan`-specific syntax considerations. Next, you will learn how to incorporate some light customization to give your slides a more personal touch. This chapter concludes with some examples of best practices for including R code in your `xaringan` presentations.

2.1 Getting started

Before we get started you must install the `xaringan` package if you have not already done so. You can install `xaringan` via CRAN by typing `install.packages("xaringan")` into the console. If you use the RStudio IDE, you can install `xaringan` by clicking the packages tab and searching for `xaringan`. If you prefer to be on the cutting edge of software development you can install the latest version of `xaringan` directly from github by typing `devtools::install_github("yihui/xaringan")` into the console, but note that this method requires that you have `devtools` installed first.

Once you have installed `xaringan` you are ready to get started. We will begin with the simplified template that is provided in the `xaringan` package. This will allow you to learn the basics of `xaringan` quickly and provide you with easy access to an example that you can refer back to if necessary. You can create a new copy of the template using the menu in the RStudio IDE. Click "File" → "New File" → "R Markdown" → "From Template" → "Ninja Presentation".[1] Once you click "OK" a new .Rmd file will automatically open in RStudio.

[1] If you are not using RStudio, that's fine too. HTML presentations generated using `xaringan` are just standard R Markdown (.Rmd) files, so you can create one from scratch using your text editor of choice and follow the examples all the same.

2.1.1 Front matter

As touched upon above, `xaringan` slides are generated from R Markdown files. This means that your presentation will have YAML front matter, like any other .Rmd file.[2] Below is the front matter of our template. If you are not using the RStudio IDE, feel free to copy the front matter and paste it into your .Rmd document.

```
---
title: "Presentation Ninja"
subtitle: " <br/>with xaringan"
author: "Yihui Xie"
date: "2016/12/12 (updated: `r Sys.Date()`)"
output:
  xaringan::moon_reader:
    lib_dir: libs
    nature:
      highlightStyle: github
      highlightLines: true
      countIncrementalSlides: false
---
```

Here you can include all of the importnat information you would typically include in the title slide of your presentation. The main difference between HTML slides generated in `xaringan` and any other R Markdown file is that the HTML output is generated using the output format `moon_reader`. The output format serves as a wrapper for remark.js, which is doing all of the heavy lifting under the hood. This information may not seem too important to most people, but it is worth mentioning because the `moon_reader` format is the only part of the front matter that is truly necessary to create your slides. In fact, if you were feeling adventurous you could create slides that *only* contained this information, i.e., `output: xaringan::moon_reader`, and delete all the other information. We will come back to remark.js in chapter 6, Advanced Topics. Now you are ready to generate your first HTML slides using `xaringan`.

2.1.2 Generating your slides

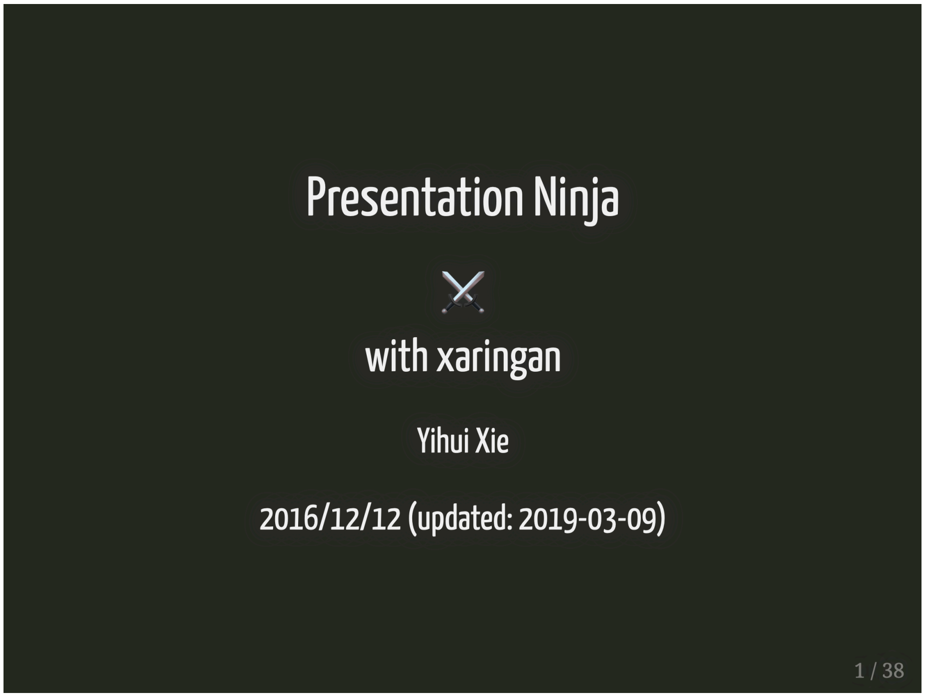
There are several methods for generating and working with slides in a `xaringan` presentation, which we will now cover. First things first, save your .Rmd file to a location on your hard drive and give your presentation file a name. If you are using RStudio, simply click the `Knit` button at the top of the application window and you will be prompted to save the .Rmd file. RStudio will render the slides after you save and they will automatically appear in the Viewer window. **Pro tip**: you can install the `infinite_moon_reader` RStudio Addin which will allow you to update your slides in real time (more on this in Chapter 6). If you

[2] You can learn the basics of R Markdown here: https://rmarkdown.rstudio.com/lesson-1.html

are not using RStudio, save your .Rmd file and then you can generate your slides from the console by typing `rmarkdown::render("your-file-name.Rmd")`.

2.1.3 Exploring a `xaringan` presentation

Before we go into the details of customizing your slides, let's explore the `xaringan` presentaion we just created. If you are using RStudio the presentation should have opened automatically in the Viewer panel. If you rendered your slides manually, double click the HTML file that was produced to open it in your web browser of choice.[3] You should see something like this:



Presentation Ninja

⚔

with xaringan

Yihui Xie

2016/12/12 (updated: 2019-03-09)

1 / 38

Figure 2.1: title_slide

Use the left and right arrow keys to navigate the slides. There are a whole host of features at your finger tips, but we will only highlight two for now. You can type the number of a given slide and press the "enter"/"return" key to navigate to that slide, i.e., "9" + "return". If you press the "p" key you will enter "Presenter mode". Cool, right? You can press the "p" key again to leave presenter mode. Type the "h" key to see a list of some of the other features. As mentioned, we will talk more about these in Chapter 6. In the next section we

[3] Note: if you are creating your slides manually, your presentation will not have much content yet. You can explore the same example slides here: https://slides.yihui.name/xaringan/

will cover the basics for editing your `xaringan` presentation.

2.2 Editing your slides

Editing a `xaringan` presentation is quite simple. If you are already familiar with Markdown syntax, you will be a `xaringan` ninja with little effort. If you are new to Markdown syntax you can cover the basics at https://rmarkdown.rstudio.com/lesson-8.html. For now, let's start with a clean slate and delete everything in our template **after** the YAML front matter. That is, delete everything in your file **after** line 13.

2.2.1 General syntax

`xaringan` shares all of the basic features of Markdown syntax. This means you can create **bold** text and *italics* by surrounding text with double or single stars, i.e., **bold** and *italics*, respectively. You can also create unordered lists:

- this
- is
- and unordered list

and ordered lists:

1. This is 1
2. This is 2
3. This is 3

like so:

```
You can also create unordered lists:

- this
- is
- and unordered list

and ordered lists:

1. This is 1
2. This is 2
3. This is 3
```

We won't cover all of the details of Markdown syntax here, but rather highlight some of the `xaringan`-specific features.

2.2.2 Creating individual slides

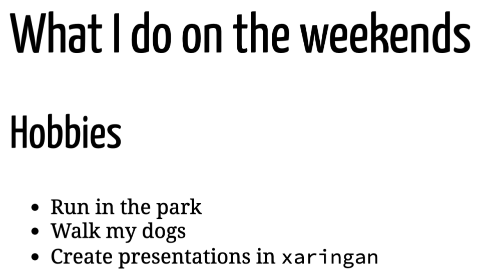
In `xaringan` you use a sequence of three hyphens, `---`, to separate slides.[4] You are not required to include any headers, but you can using standard Markdown syntax, i.e., hashtags. Thus, we could create a simple slide including a header, a subsection, and a list by including the following code after the YAML front matter:[5]

```
# What I do on the weekends

## Hobbies

- Run in the park
- Walk my dogs
- Create presentations in `xaringan`
```

Feel free to copy this into your file and click "Knit". The output should render like this:



What I do on the weekends

Hobbies

- Run in the park
- Walk my dogs
- Create presentations in xaringan

Figure 2.2: basics-slide-example

2.2.3 Incremental slides

It is often useful in a presentation to incrementally show information. This is easily done in `xaringan` using two hyphens, `--`. Careful, don't confuse this with the three hyphen slide breaks! You can incrementally reveal pretty much any type of inforamtion you put into a slide. Let's add some examples to our

[4] Note that this is slightly different from ioSlides where hastags, ##, are used to dilineate where a new slide begins.

[5] This implies that the final three hyphens of the YAML front matter indicate the end of the title slide and the beginning of slide #2.

presentation by creating a new slide. Don't forget, to create the new slide we need to add `---`. In our case, this would be after the last line of list we created in the previous example. For the sake of completeness I will include the code from both slides below.

```
# What I do on the weekends

## Hobbies

- Run in the park
- Walk my dogs
- Create presentations in `xaringan`

---

# Incremental slides

### Sentence example

This sentence
--
is displayed
--
incrementally.
```

If you copy this example and Knit your slides you will notice notice that the sentence is rendered incrementally on a single line even though it is written in on separate lines. This is because the double hyphen, `--`, has to be on its own line with no spaces before or after in order to work properly. We can also use this technique to reveal items of a list incrementally:

```
# Incremental slides

### Non-incremental sentence example

This sentence is no longer displayed incrementally.

### Incremental list example

- An incremental list
--

- Is useful
--

- Sometimes
```

2.2.4 Multi-column slides

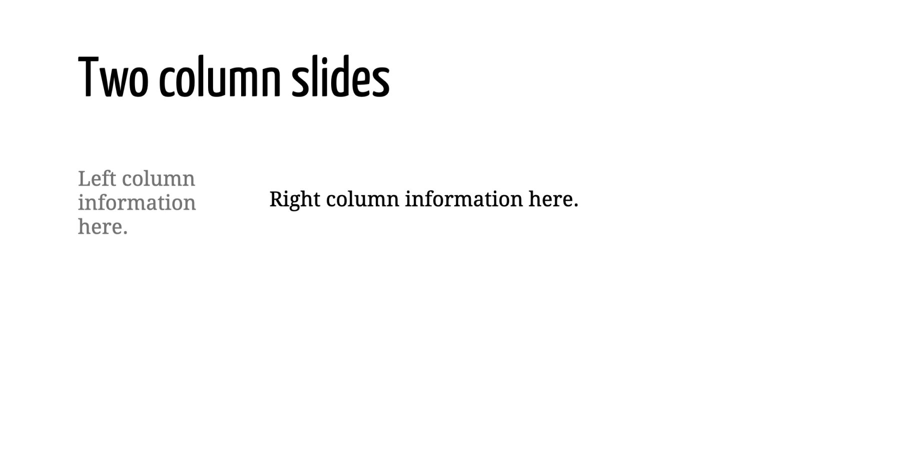
Another common feature of presentations is to display content in two columns. This is easibly acheived in `xaringan` using a bit of remark.js syntax. You will recognize the remark.js syntax because it begins with a period and and opening bracket, `.[`, and end with a closing bracket, `]`. Specifically, there are two options for creating two-column layouts. The first option is to use `.left-column[]` and `.right-column[]`. For example:

```
# Two column slides

.left-column[
Left column information here.
]

.right-column[
Right column information here.
]
```

would render like this:



Two column slides

Left column information here.

Right column information here.

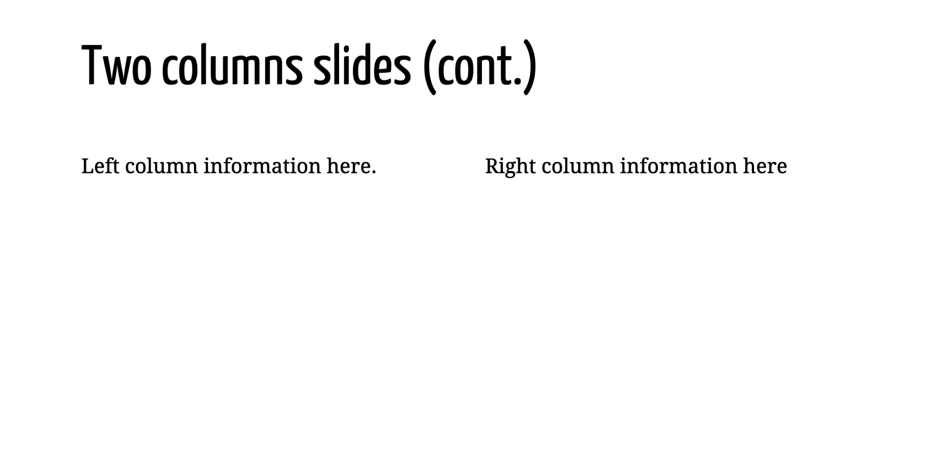
Figure 2.3: basic-two-column-1

As you can see, this divides the left column into approxametely 1/3 of the slide width and the right column contains the remaining 2/3. This may be most useful when you need show a list on one side and need more space for content on the other. Another option is to divide the slide equally down the middle. You can achieve this by using `.pull-left[]` and `.pull-right[]`. For example:

```
# Two columns slides (cont.)

.pull-left[
Left column information here.
]

.pull-right[
Right column information here
]
```



Two columns slides (cont.)

Left column information here.

Right column information here

Figure 2.4: basic-two-column-2

2.2.5 math

2.3 Making it yours

2.3.1 Images, gifs

- knitr
- background images

2.3.2 Templates

- names(xaringan:::list_css())

2.3.3 HTML/CSS/JS basics

- iframes
  - include other websites, shiny apps
- basic customization (CSS, JS)
  - colors, counters, tweets

2.4 Incorporating R code

- analyses, plots and tables (some html widgets)
- inline
- knitr chunks
- code highlighting (`*`, `{{}}`, and `#<<`; `highlight.output`)

2.5 Summary

Chapter 3

remark.js

Ideas so far:

Features of the remark.js. Can draw out from wiki: https://github.com/gnab/remark/wiki

Chapter 4

Features

Ideas so far:

- `xaringan` specific features not in remark.js

Chapter 5

Publishing

Ideas so far:

- Where to host (netlify and github pages)
- Offline
- Any chance that RPubs to support xaringan in the future? Nice if tag feature is added to RPubs too. Or connect to some blogdown site with something similar to https://www.overleaf.com/latex/templates

Chapter 6

Advanced Topics

So advanced that you need to train for weeks to know what it is.

Chapter 7

Ideas

Some inspirations and theme showcase?