Introduction to Rmarkdown/Quarto for Reproducibility

SCO-SOC Workshop 2022

Steffi LaZerte

Getting Started

Introduction

Today learning to create static HTML reports from R code

(but can also create websites, and presentations–like this one!)

Why?

- Keep track of your code *and* results

- Share your work

- Ensure reproducibility

- Be nice to your future self (*What did I do again? What were the results?*)

Okay, what kind of report?

For example…

```
## Setup
This is my **great** study.... I used these packages:

```{r}
library(tidyverse)
```


Loading data
These are the datasets I used

```{r}
my_data <- read_csv("https://raw.githubusercontent.com/steffilazerte/NRI_7350/main/data/
my_data
```


This is what it looks like

```{r}
#| fig.width = 6
ggplot(data = my_data, aes(x = urbanization, y = songs)) +
  geom_point()
```
```

Becomes…

Setup

This is my **great** study…. I used these packages:

```
1  library(tidyverse)
```

```
── Attaching packages ──────────────────────────────── tidyverse 1.3.2 ──
✔ ggplot2 3.3.6      ✔ purrr   0.3.4
✔ tibble  3.1.7      ✔ dplyr   1.0.9
✔ tidyr   1.2.0      ✔ stringr 1.4.0
✔ readr   2.1.2      ✔ forcats 0.5.1
── Conflicts ──────────────────────────────── tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
```

Loading data

These are the datasets I used

```
1  my_data <- read_csv("https://raw.githubusercontent.com/steffilazerte/NRI_7350/main/data/chorus.csv")
2  my_data
```

```
# A tibble: 51 × 3
   urbanization songs calls
          <dbl> <dbl> <dbl>
 1        0.794     0   136
 2        0.890    60    12
 3       -1.85     55    66
 4       -1.85     22   115
```

For another example…

```
### Visual of Thresholds Calculations

> - Pink ribbon = 99% Confidence interval of latitudes predicted from GAM
> - Black lines in the ribbon are the upper and lower limit, the middle
line is the predicted latitude (from GAM model)
> - Transparent blue rectangles indicate the date ranges used to establish
the latitudes just after and just before migration.
> - Blue horizontal lines represents the latitude threshold for spring
    migration (begin/end)
> - Orange horzontal lines represents the latitude threshold for fall
    migration (begin/end)

```{r}
#| fig.asp: 1
#| fig.width: 15
wrap_plots(g) + plot_layout(guides = "collect", nrow = 1)
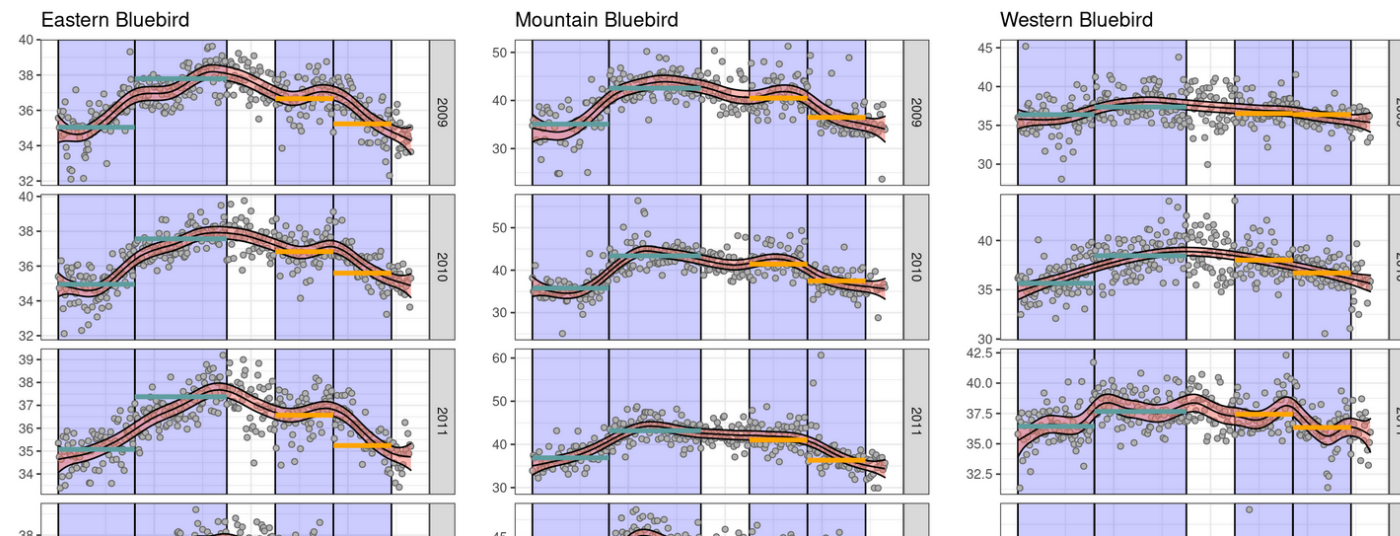```
```

(Plus a bunch of other options)

Becomes...

Visual of Thresholds Calculations

- Pink ribbon = 99% Confidence interval of latitudes predicted from GAM
- Black lines in the ribbon are the upper and lower limit, the middle line is the predicted latitude (from GAM model)
- Transparent blue rectangles indicate the date ranges used to establish the latitudes just after and just before migration.
- Blue horizontal lines represents the latitude threshold for spring migration (begin/end)
- Orange horzontal lines represents the latitude threshold for fall migration (begin/end)

```
wrap_plots(g) + plot_layout(guides = "collect", nrow = 1)
```

Wait a minute…

That doesn't look like an R Script…

```
## Setup
This is my **great** study.... I used these packages:

```{r}
library(tidyverse)
```

Loading data
These are the datasets I used

```{r}
my_data <- read_csv("https://raw.githubusercontent.com/steffilaze
my_data
```

This is what it looks like

```{r}
#| fig.width = 6
ggplot(data = my_data, aes(x = urbanization, y = songs)) +
  geom_point()
```
```

**Four things going on…**

1. R code

2. Markdown

3. R code fences (define code *chunks*)

4. YAML *chunk* options

This is actually not an **.R** script…
it's an **Rmarkdown (.Rmd)** or
**Quarto (.qmd)** document!

# Quick start

- File > New File > Quarto Document (or R Markdown, if you prefer)

- Add details, click "Create"

- Click "Render" button in the top panel (Quarto)

  - or "Knit" button (R Markdown)

Demo

# Your Turn

Using this RStudio template, add in some code from your own scripts and render it.

Keep it relatively simple for now 😉

> What just happened? What are all these things?
> **Rmarkdown? Markdown? Quarto?**

# Terminology

**R & RStudio**

- Both are programs

- R is the programming language/envrionment

- RStudio is an IDE (integrated development environment)

# Terminology

## Markdown

- A text markup language

- Files are .md

**For example, the following…**

```
1    ### My heading
2
3    **Hi!** This is in *italics*
4
5    A [link](https://cran.r-project.org/) to R
```

Becomes…

My heading

Hi! This is in *italics*

A link to R

Terminology

RMarkdown, Quarto, knitr, and Pandoc

- **RMarkdown(. Rmd**) and **Quarto (. qmd**) files are a *mix* of Markdown *and* R code

- **knitr** is an R package which evaluates R code and returns the output as a Markdown file

- **Pandoc** is a separate (independent) program that converts Markdown to a variety of formats

Rmarkdown vs. Quarto

Quarto (. qmd) is the *next generation* of Rmarkdown (. Rmd). You can still use Rmarkdown (it's not going anywhere), but Quarto is much newer and more powerful.

Terminology

YAML, HTML, CSS

- **YAML** is a language for specifying metadata

    - Used for specifying document options and *chunk* options

- **HTML** is a language for making websites

    - Can be used directly in `.qmd`/`.Rmd` files if you plan to output to HTML

    - E.g., can use `<br>` for a line **br**eak

- **CSS** is a language for *styling* websites

    - Can be used to apply custom styles to documents

Some options

Document level options - YAML block

```
1  ---
2  title: "My great analysis"
3  format: html
4  date: today
5  toc: true
6  code-fold: true
7  ---
```

- `date: today` to include today's date

- `toc: true` to include a table of contents

- `code-fold: true` to hide code (with option to show)

Note: These are Quarto options! RMarkdown has similar ones, but they may be slightly different. E.g., `format: html_document` in RMarkdown.

Some options

Chunk level options - YAML notation

```{r}
#| fig.width: 10
#| fig.asp: 0.5
#| fig.alt: |
#|   A scatterplot in black and white showing degree of
#|   urbanization on the x-axis and number of songs on
#|   the y-axis with no appreciable pattern in the data.
#| fig.cap: |
#|   The relationship between urbanization and the number
#|   of songs in mountain chickadee dawn choruses.

ggplot(data = my_data, aes(x = urbanization, y = songs)) +
  geom_point()
```

- `fig.width` width of figure in inches
- `fig.asp` aspect of the figure (1 = square) (i.e. height = width * aspect)
- `fig.alt` **Accessibility** Alt text for screen readers helping those who can't see the figure (should be descriptive, not the same as a caption)
- `fig.cap` Figure caption

Gives…

```
1  ggplot(data = my_data, aes(x = urbanization, y = songs)) +
2    geom_point()
```
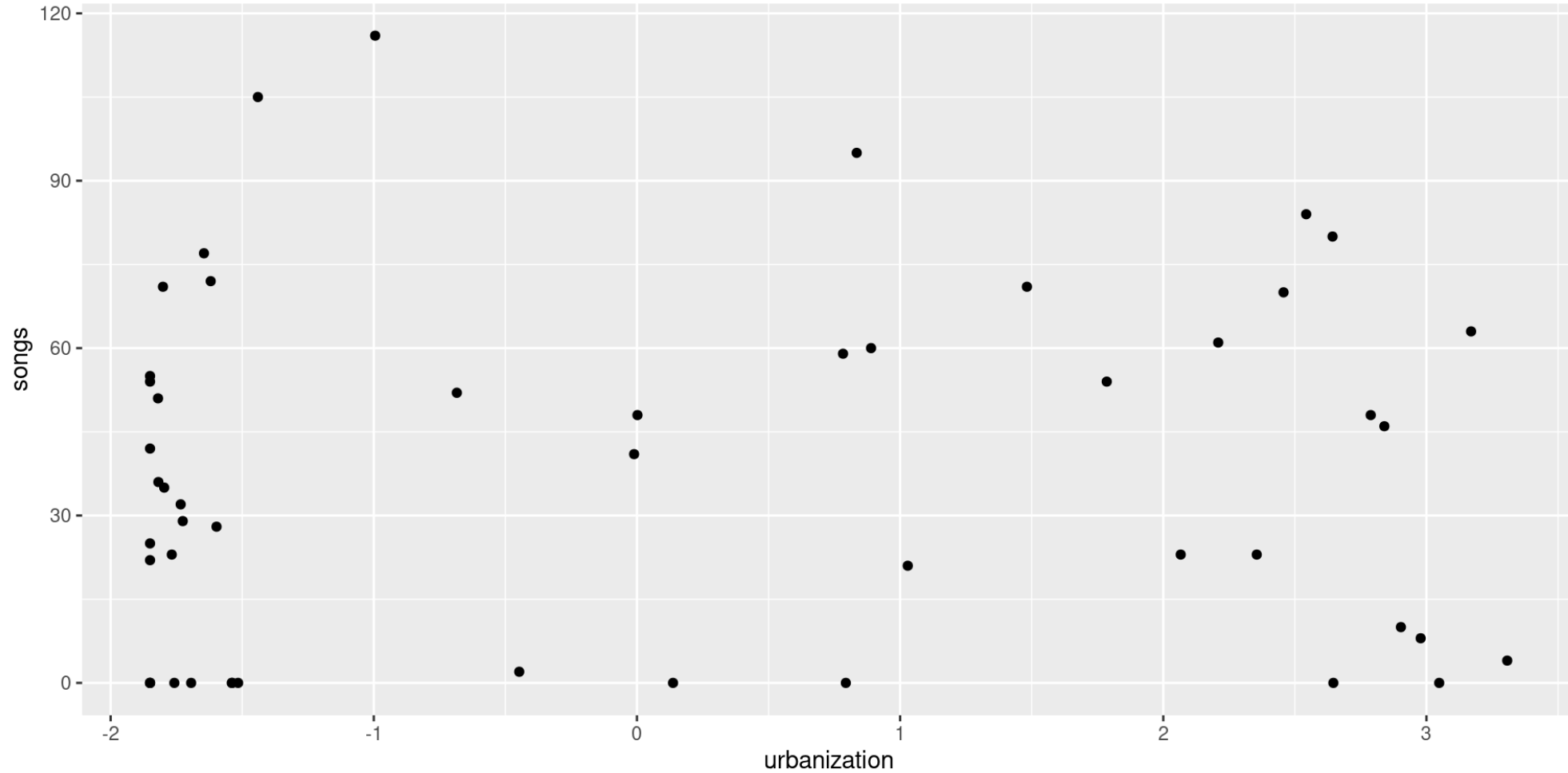

The relationship between urbanization and the number of songs in mountain chickadee dawn choruses.

Enhancing reproducibility

- Make your publication figures in reports

- Date your reports (`my_analysis_2022-09-08.html`)

- Include info on packages used (because you're going to cite them… right? **RIGHT?**)

  - `devtools::session_info()`

  - `report::report_packages()`

  - `report::cite_packages()`

- Embed data directly (for smaller datasets) using DT package

```
1  DT::datatable(mtcars, extensions = 'Buttons',
2              options = list(dom = 'Bfrtip', buttons = c('csv', 'excel')))
```

And seriously, **cite the packages** 😁

Your Turn

Use the more advanced template (`example.qmd`) to create a reproducible report of your analysis.

Consider the options we learned about. Anything you'd like to add?

Some Final Thoughts

Rendering vs. Spinning

Rendering (Render/Knit button)

`.Rmd`/`.qmd` → `.md` → HTML

- Good for lots of text
- Better option control
- Use ` ```{r} ` and ` ``` `to define code blocks

Spinning (Knit button)

`.R` → `.md` → HTML

- Easier to code
- Use `#'` to define markdown
- Use `#+` to define chunk options

Quarto notation (slightly different from Rmarkdown) *only* for `.qmd` documents so far

Rendering vs. Spinning

Rendering (Render/Knit button)

```
## Setup
This is my **great** study.... I used these packages:

```{r}
library(tidyverse)
```

Loading data
These are the datasets I used

```{r}
my_data <- read_csv("https://raw.githubusercontent.co
my_data
```


This is what it looks like

```{r}
#| fig.width = 6
ggplot(data = my_data, aes(x = urbanization, y = songs
  geom_point()
```
```

## Spinning (Knit button)

```
 1  #' ## Setup
 2  #' This is my **great** study.... I used these pad
 3
 4  library(tidyverse)
 5
 6  #' ## Loading data
 7  #' These are the datasets I used
 8
 9  my_data <- read_csv("https://raw.githubusercontent
10  my_data
11
12  #' This is what it looks like
13
14  #+ fig.width = 6
15  ggplot(data = my_data, aes(x = urbanization, y = s
16    geom_point()
```

Relative locations

If you use nested folders in your work, you'll want to use the here package to ensure all the file locations are consistent

```r
library(here)
library(tidyverse)

my_data <- read_csv(here("Data/my_data.csv"))
```

Resources

Online References

- Quarto Documentation

- Openscapes' Quarto Tutorial

- RStudio's Welcome to Quarto Workshop! (video)

- We don't talk about Quarto (blog post)

- A Quarto tip a day (blog)

- R Markdown Documentation

- R Markdown: The Definitive Guide (online book)

- RStudio > Help > Markdown Quick Reference

- RStudio > Help > Cheat Sheets > Rmarkdown Cheat Sheet

- RStudio > Help > Cheat Sheets > Rmarkdown Reference Guide

Examples

- This presentation

- Quarto Gallery

- R Markdown Gallery