Introduction to Quarto for Reproducibility

UNBC Applied Analysis Hub

steffilazerte
@steffilazerte@fosstodon.org
steffilazerte.ca

Dr. Steffi LaZerte
Analysis and Data Tools for Science

Compiled: 2025-09-23

Preamble

Online workshops can be challenging

Online workshops can be challenging

Consider keeping your video on (if possible)

- Kids? Pets? Spouses? No problem
- But ultimately, you need be comfortable! (and you absolutely have the right to privacy)

Online workshops can be challenging

Consider keeping your video on (if possible)

- Kids? Pets? Spouses? No problem
- But ultimately, you need be comfortable! (and you absolutely have the right to privacy)

Interrupt me!

- Generally keep yourself muted but un-mute anytime to ask questions

Online workshops can be challenging

Consider keeping your video on (if possible)

- Kids? Pets? Spouses? No problem
- But ultimately, you need be comfortable! (and you absolutely have the right to privacy)

Interrupt me!

- Generally keep yourself muted but un-mute anytime to ask questions

Ask Questions!

- Group trouble-shooting is really valuable
- If you have a problem, others may also (or may have it in the future)

Online workshops can be challenging

Consider keeping your video on (if possible)

- Kids? Pets? Spouses? No problem
- But ultimately, you need be comfortable! (and you absolutely have the right to privacy)

Interrupt me!

- Generally keep yourself muted but un-mute anytime to ask questions

Ask Questions!

- Group trouble-shooting is really valuable
- If you have a problem, others may also (or may have it in the future)

Screen-sharing

- I may ask you to share your screen with the group (feel free to decline)
- For privacy, close your email etc. Or just share your RStudio window

Introductions

Me and my creatures

My garden

What about you?

- Name
- Background (Role, Area of study, etc.)
- Familiarity with R or Programming
- Creatures (furry, feathery, scaley, green or otherwise)?

Getting Started

Today we’re learning to create static HTML reports from R code

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

Getting Started

Today we’re learning to create static HTML reports from R code

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

Why?

Getting Started

Today we’re learning to create static HTML reports from R code

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

Why?

- Keep track of your code *and* results

Getting Started

Today we’re learning to create static HTML reports from R code

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

Why?

- Keep track of your code *and* results
- Share your work

Getting Started

Today we’re learning to create static HTML reports from R code

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

Why?

- Keep track of your code *and* results
- Share your work
- Ensure reproducibility

Getting Started

Today we're learning to create static HTML reports from R code

(but can also create websites, pdfs, 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?*)

Getting Started

Today we're learning to create static HTML reports from R code

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

Why?

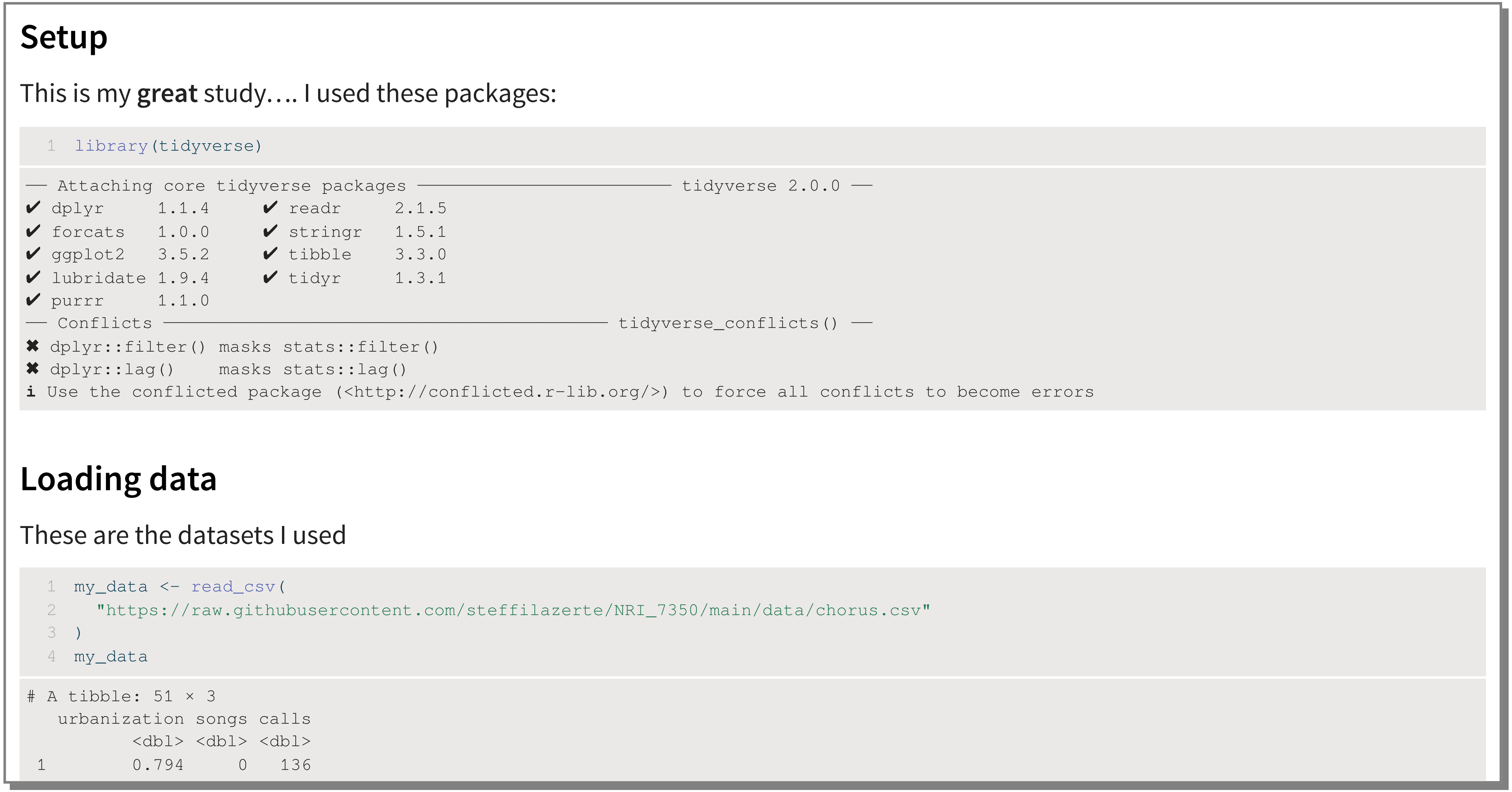
- Keep track of your code *and* results
- Share your work
- Ensure reproducibility
- Be nice to your future self (*What did I do again? What were the results?*)

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/chorus.csv")
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:

```r
library(tidyverse)
```

```
── Attaching core tidyverse packages ────────────────────── tidyverse 2.0.0 ──
✔ dplyr     1.1.4     ✔ readr     2.1.5
✔ forcats   1.0.0     ✔ stringr   1.5.1
✔ ggplot2   3.5.2     ✔ tibble    3.3.0
✔ lubridate 1.9.4     ✔ tidyr     1.3.1
✔ purrr     1.1.0
── Conflicts ──────────────────────────────────────── tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

Loading data

These are the datasets I used

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

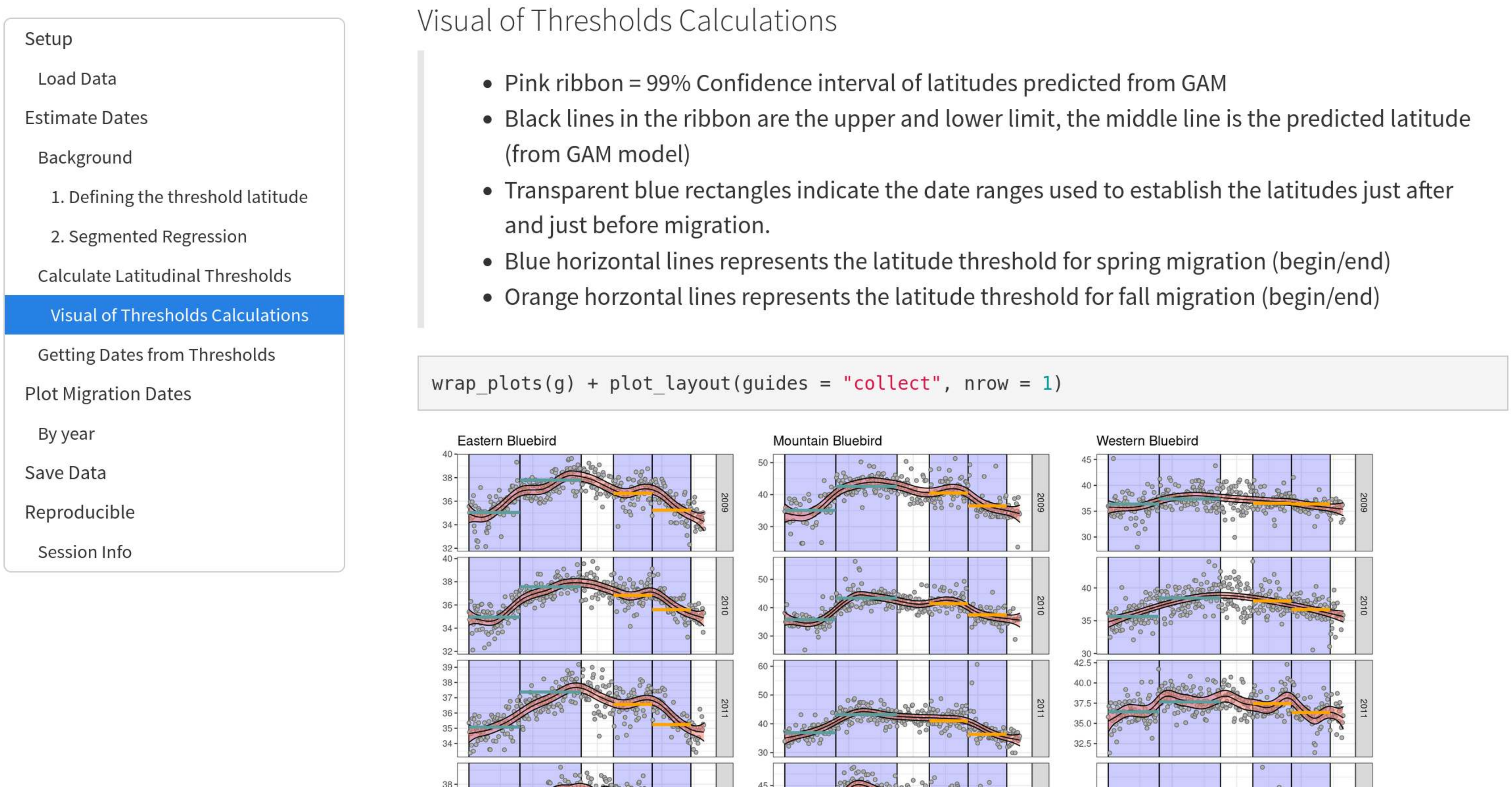
```
# A tibble: 51 × 3
  urbanization songs calls
         <dbl> <dbl> <dbl>
1        0.794     0   136
```

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…

Not 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/steffila
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. R code fences (define code *chunks*)
3. Markdown
4. YAML *chunk* options
```

# Not 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/steffila
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. R code fences (define code *chunks*)
3. Markdown
4. YAML *chunk* options

Not 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/steffila
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. R code fences (define code *chunks*)
3. Markdown
4. YAML *chunk* options
```

Not 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/steffila
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. R code fences (define code *chunks*)
3. Markdown
4. YAML *chunk* options
```

# Not 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/steffila
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. R code fences (define code *chunks*)
3. Markdown
4. YAML *chunk* options
```

Not 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/steffila
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. R code fences (define code *chunks*)
3. Markdown
4. YAML *chunk* options

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

# Quick start

- File > New Project
- 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 😉

# 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?
**R Markdown? Markdown? Quarto? YAML** 😱

# Terminology

## R & RStudio

- Both are programs
- R is the programming language/envrionment
- RStudio is an IDE (integrated development environment)

R

RStudio

# Terminology

## Markdown

- A text markup language
- Files are `.md`

**For example, the following…**

```
### My heading

**Hi!** This is in *italics*

A [link](https://cran.r-project.org/) to R
```

Becomes…

My heading

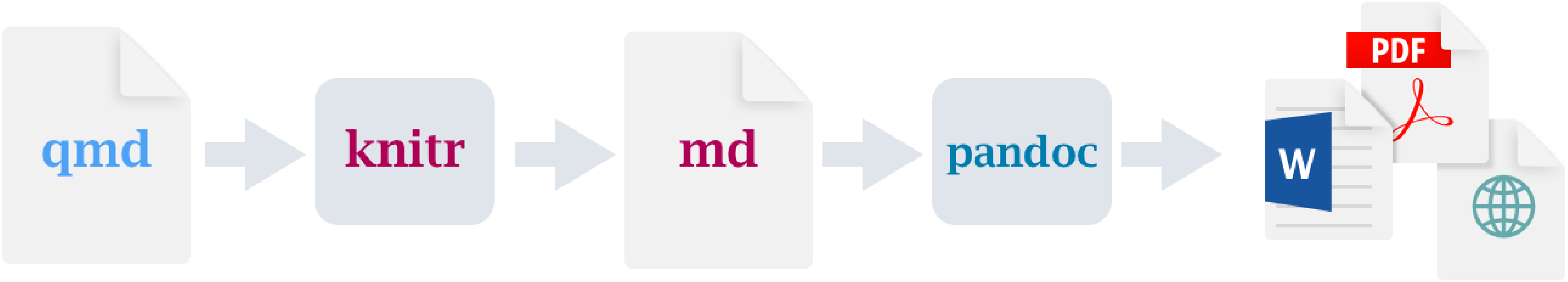
Hi! This is in *italics*

A link to R

Terminology

R Markdown, Quarto, knitr, and Pandoc

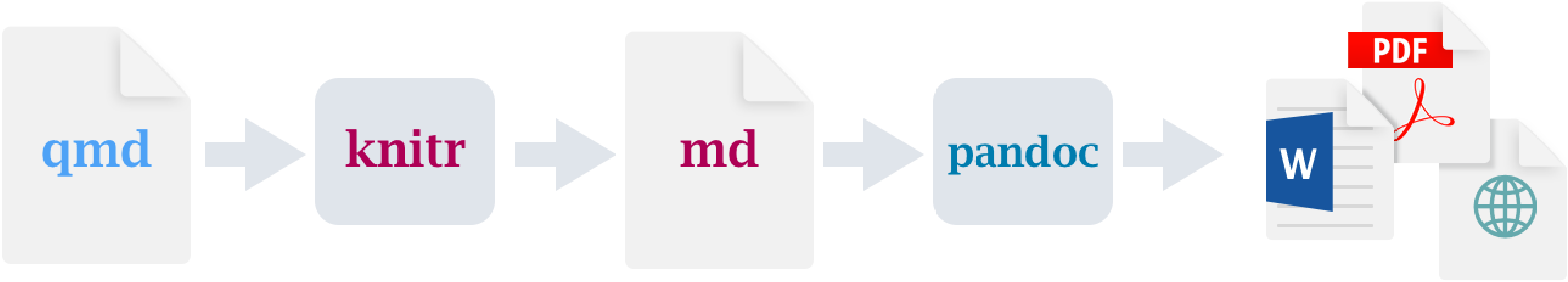
- **R Markdown**(`.Rmd`) and **Quarto** (`.qmd`) files are a *mix* of Markdown *and* R code
- **knitr** is an R package which evaluates R code and returns the output as a Markdown file
- **Pandoc** is a separate (independent) program that converts Markdown to a variety of formats



Terminology

R Markdown, Quarto, knitr, and Pandoc

- **R Markdown**(`.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



R Markdown vs. Quarto

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

Terminology

YAML, HTML, CSS/SCSS

Terminology

YAML, HTML, CSS/SCSS

- **YAML** is a language for specifying (meta)data
  - Used for specifying document options/configuration and *chunk* options

Terminology

YAML, HTML, CSS/SCSS

- **YAML** is a language for specifying (meta)data
  - Used for specifying document options/configuration 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

Terminology

YAML, HTML, CSS/SCSS

- **YAML** is a language for specifying (meta)data
  - Used for specifying document options/configuration 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
  - **SCSS** is CSS with superpowers

Some options

Document level options - YAML block

```yaml
---
title: "My great analysis"
format: html
date: today
toc: true
code-fold: true
---
```

- `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! R Markdown has similar ones, but they may be slightly different. E.g., `format: html_document` in R Markdown.

Some options

Chunk level options - YAML notation

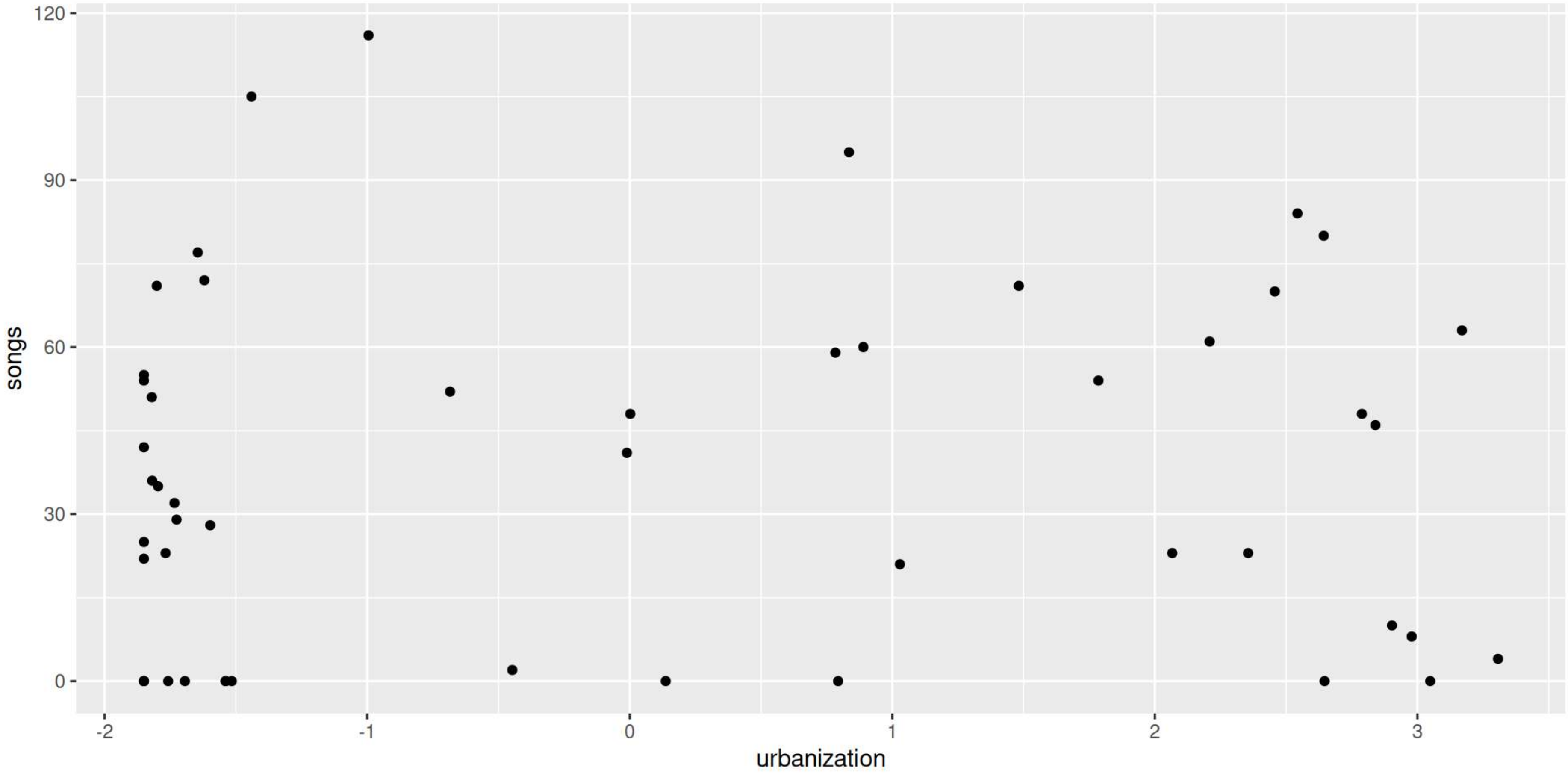
~~~
```{r}
#| fig-width: 10
#| fig-asp: 0.5
#| fig-alt: |
#|   A scatterplot in black and white showing degree of
#|   urbanization on the x-axis and number of songs on
#|   the y-axis with no appreciable pattern in the data.
#| fig-cap: |
#|   The relationship between urbanization and the number
#|   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` **Alt**ernative text for screen readers helping those who can't see the figure (should be descriptive, not the same as a caption)[1]
- `fig-cap` Figure caption

[1] More alt text resources: when to include; how to write; the importance of context

Gives...

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



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

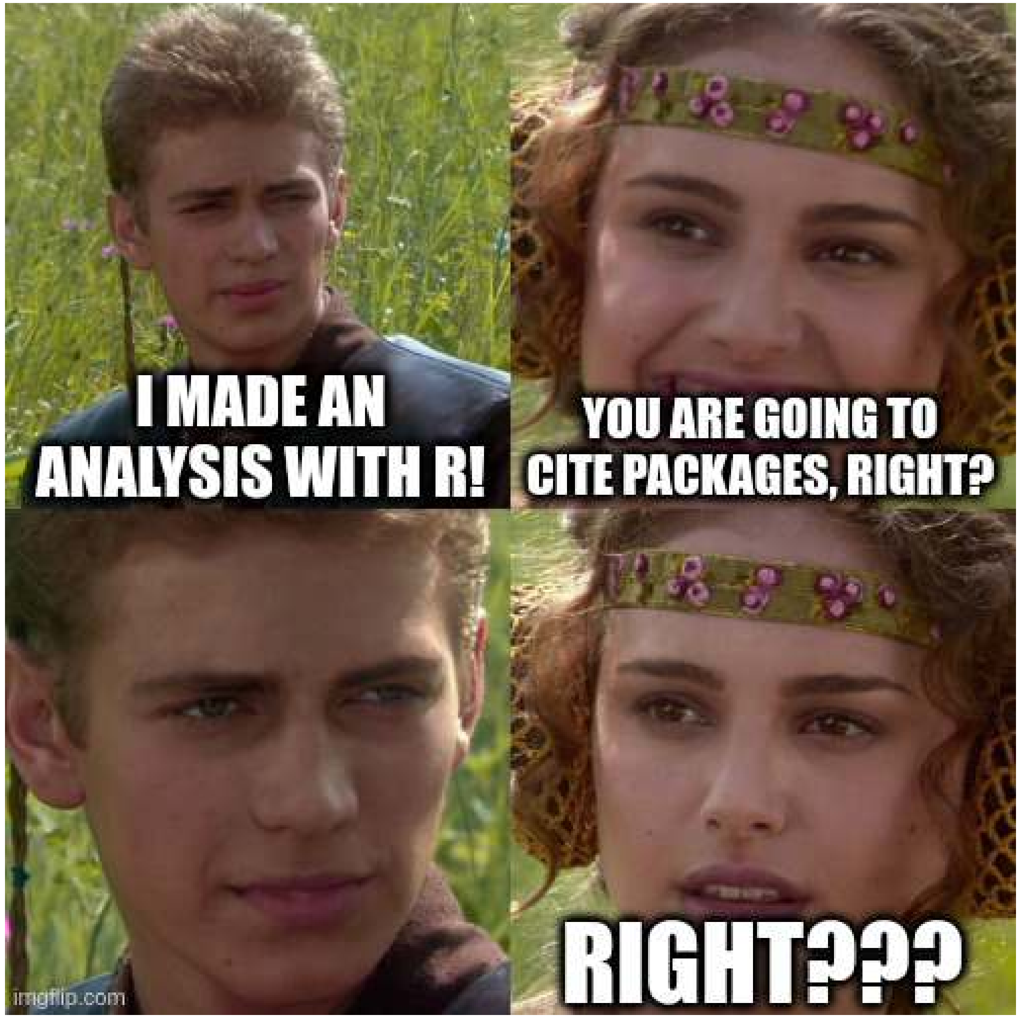
Enhancing reproducibility

- Create your publication figures through 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

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

Cite the Packages!

Seriously, cite **the packages** 😁


I MADE AN ANALYSIS WITH R!
YOU ARE GOING TO CITE PACKAGES, RIGHT?
RIGHT???
imgflip.com

Your Turn

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

Consider the options we learned

Anything you'd like to add?

Consider `code-fold: true` or `toc: false`

Some Final Thoughts

Rendering vs. Spinning

Rendering (Render/Knit button)

`.Rmd`/`.qmd` → `.md` → HTML

Spinning (Knit button)

`.R` → `.md` → HTML

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
  - Use Rmarkdown option style
  - i.e., `error=FALSE` not `error: false`

Rendering vs. Spinning

See `example_spin.R`

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.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()
```
~~~

Or render with:

```
quarto::quarto_render(
  input = "example.qmd",
  output_file = paste0("example_", Sys.Date(), ".html"))
```

Spinning (Knit button)

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

library(tidyverse)

#' ## Loading data
#' These are the datasets I used

my_data <- read_csv("https://raw.githubusercontent.com/steffilaze
my_data

#' This is what it looks like

#+ fig-width = 6
ggplot(data = my_data, aes(x = urbanization, y = songs)) +
  geom_point()
```

Or spin/render with:

```
knitr::spin("example_spin.R", knit = FALSE)
quarto::quarto_render(
  input = "example_spin.Rmd",
  output_file = paste0("example_spin_", Sys.Date(), ".html"))
```

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"))
```

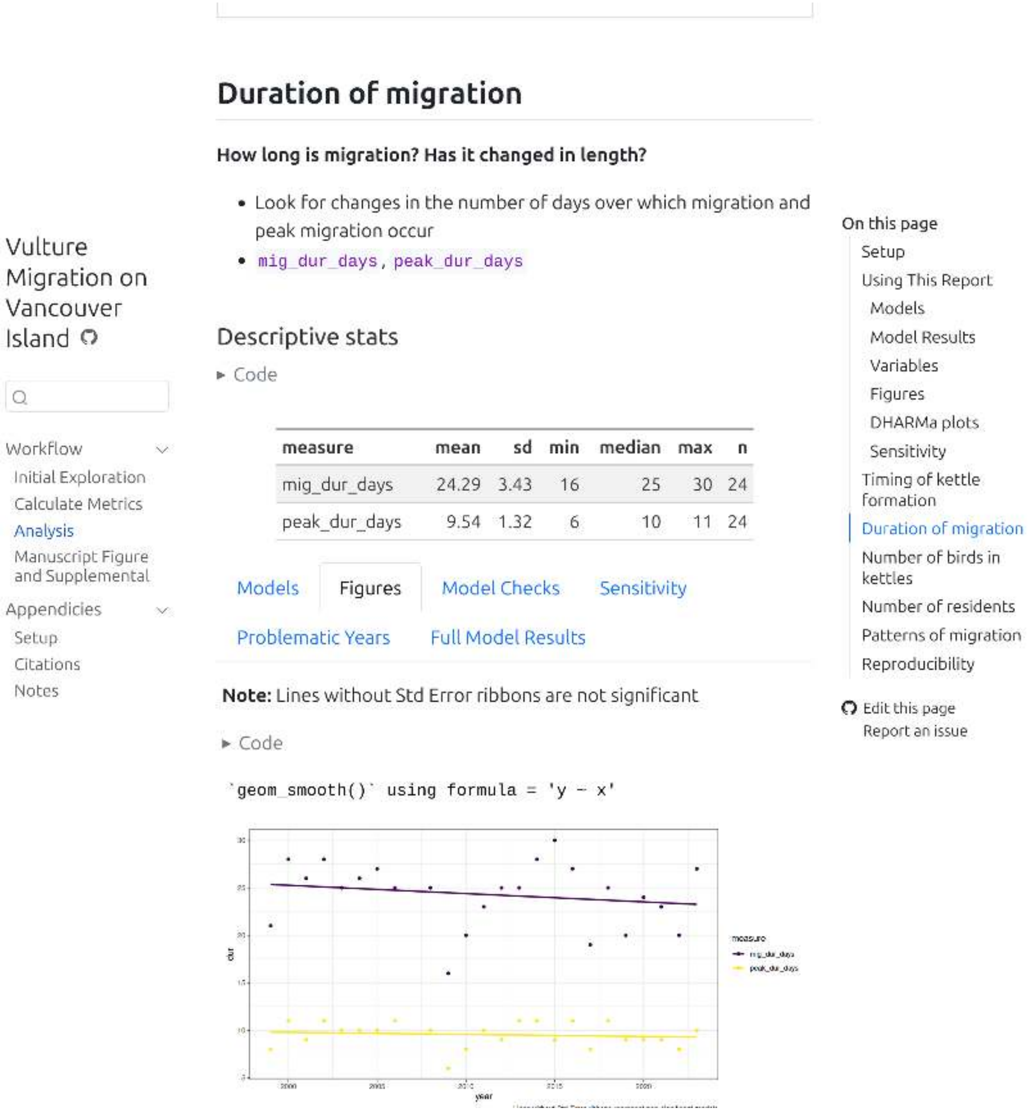
Artwork by @allison_horst

Taking it to the next level

Analysis Websites

(Quarto Docs for Websites)

Vulture Migration (GitHub | Website)



Vulture Migration on Vancouver Island

Workflow
- Initial Exploration
- Calculate Metrics
- Analysis
- Manuscript Figure and Supplemental

Appendicies
- Setup
- Citations
- Notes

Duration of migration

How long is migration? Has it changed in length?

- Look for changes in the number of days over which migration and peak migration occur
- `mig_dur_days`, `peak_dur_days`

Descriptive stats

▸ Code

measure	mean	sd	min	median	max	n
mig_dur_days	24.29	3.43	16	25	30	24
peak_dur_days	9.54	1.32	6	10	11	24

Models | Figures | Model Checks | Sensitivity | Problematic Years | Full Model Results

Note: Lines without Std Error ribbons are not significant

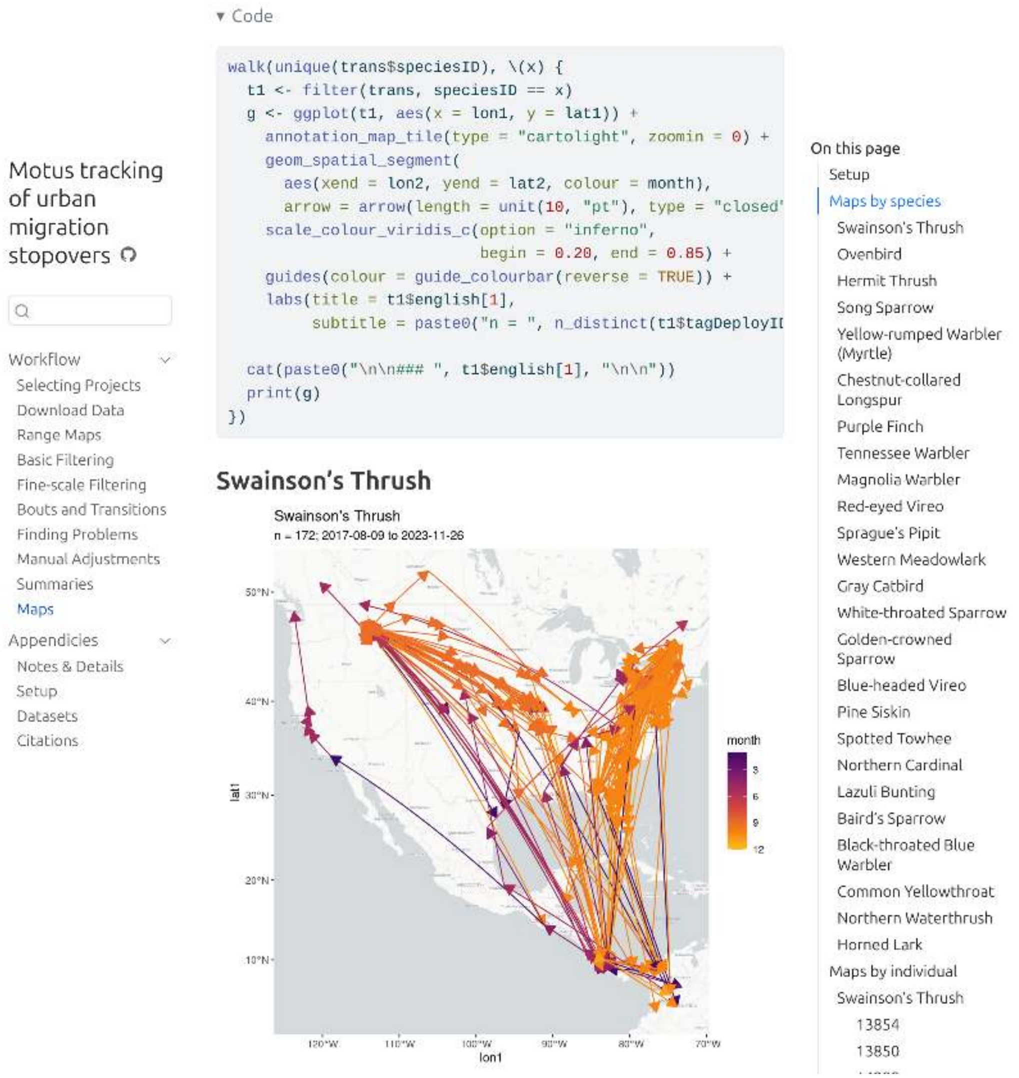
▸ Code

```
`geom_smooth()` using formula = 'y ~ x'
```

On this page
- Setup
- Using This Report
  - Models
  - Model Results
  - Variables
  - Figures
  - DHARMa plots
  - Sensitivity
- Timing of kettle formation
- Duration of migration
- Number of birds in kettles
- Number of residents
- Patterns of migration
- Reproducibility

Edit this page
Report an issue

Urban Migration Stopovers (GitHub | Website)



Motus tracking of urban migration stopovers

Workflow
- Selecting Projects
- Download Data
- Range Maps
- Basic Filtering
- Fine-scale Filtering
- Bouts and Transitions
- Finding Problems
- Manual Adjustments
- Summaries
- Maps

Appendicies
- Notes & Details
- Setup
- Datasets
- Citations

▾ Code

```
walk(unique(trans$speciesID), \(x) {
  t1 <- filter(trans, speciesID == x)
  g <- ggplot(t1, aes(x = lon1, y = lat1)) +
    annotation_map_tile(type = "cartolight", zoomin = 0) +
    geom_spatial_segment(
      aes(xend = lon2, yend = lat2, colour = month),
      arrow = arrow(length = unit(10, "pt"), type = "closed'
    scale_colour_viridis_c(option = "inferno",
                           begin = 0.20, end = 0.85) +
    guides(colour = guide_colourbar(reverse = TRUE)) +
    labs(title = t1$english[1],
         subtitle = paste0("n = ", n_distinct(t1$tagDeployII

  cat(paste0("\n\n### ", t1$english[1], "\n\n"))
  print(g)
})
```

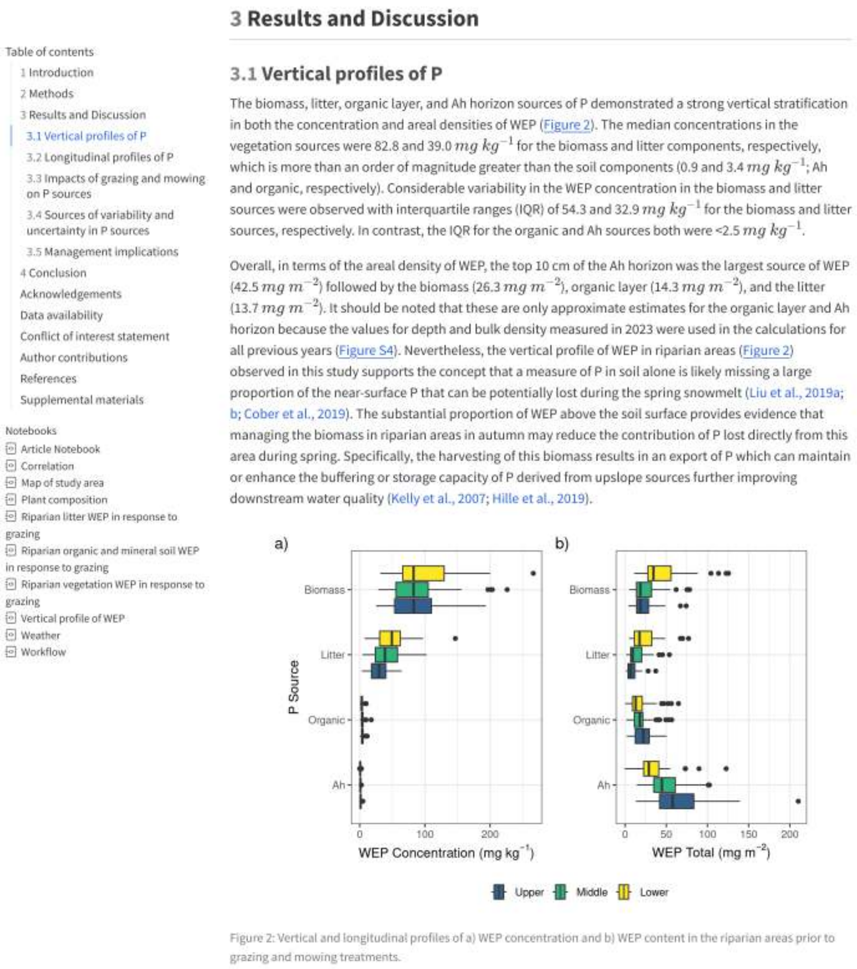
Swainson's Thrush

On this page
- Setup
- Maps by species
  - Swainson's Thrush
  - Ovenbird
  - Hermit Thrush
  - Song Sparrow
  - Yellow-rumped Warbler (Myrtle)
  - Chestnut-collared Longspur
  - Purple Finch
  - Tennessee Warbler
  - Magnolia Warbler
  - Red-eyed Vireo
  - Sprague's Pipit
  - Western Meadowlark
  - Gray Catbird
  - White-throated Sparrow
  - Golden-crowned Sparrow
  - Blue-headed Vireo
  - Pine Siskin
  - Spotted Towhee
  - Northern Cardinal
  - Lazuli Bunting
  - Baird's Sparrow
  - Black-throated Blue Warbler
  - Common Yellowthroat
  - Northern Waterthrush
  - Horned Lark
- Maps by individual
  - Swainson's Thrush
    - 13854
    - 13850

Manuscript Websites

(Quarto Docs for Manuscripts)

Effects of Riparian Grazing on Distinct Water-Extractable Phosphorus Sources (GitHub | Website)



Notebooks

- Article Notebook
- Correlation
- Map of study area
- Plant composition
- Riparian litter WEP in response to grazing
- Riparian organic and mineral soil WEP in response to grazing
- Riparian vegetation WEP in response to grazing
- Vertical profile of WEP
- Weather
- Workflow

3 Results and Discussion

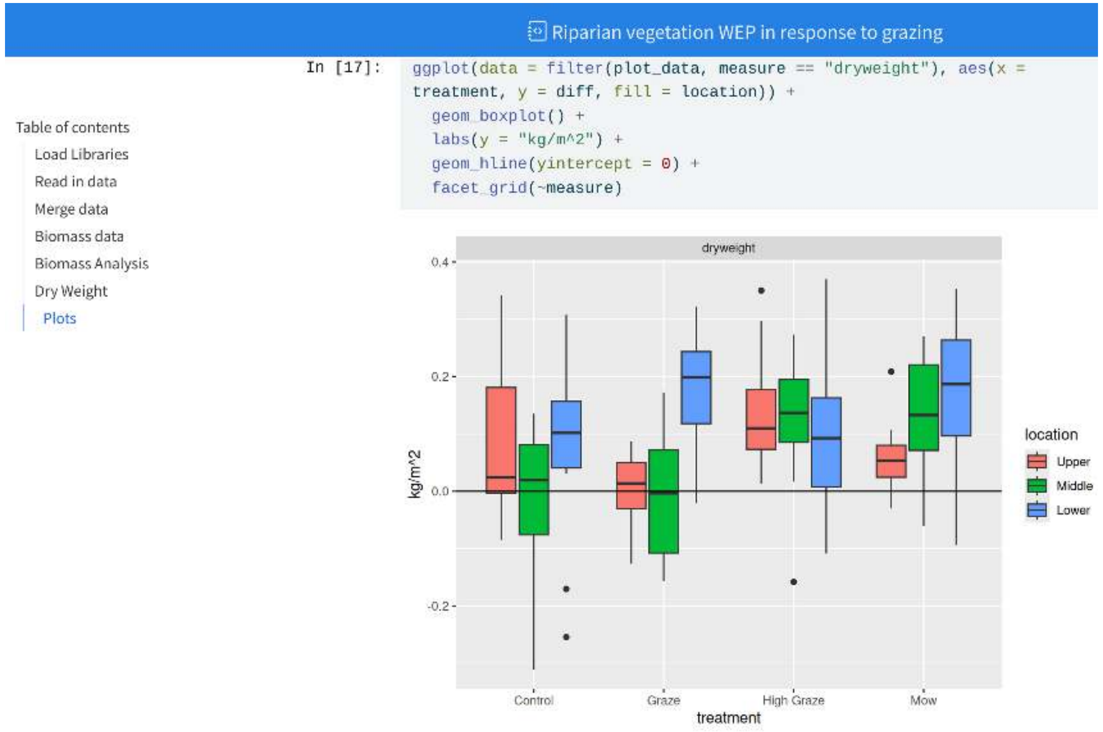
3.1 Vertical profiles of P

The biomass, litter, organic layer, and Ah horizon sources of P demonstrated a strong vertical stratification in both the concentration and areal densities of WEP (Figure 2). The median concentrations in the vegetation sources were 82.8 and 39.0 $mg\ kg^{-1}$ for the biomass and litter components, respectively, which is more than an order of magnitude greater than the soil components (0.9 and 3.4 $mg\ kg^{-1}$; Ah and organic, respectively). Considerable variability in the WEP concentration in the biomass and litter sources were observed with interquartile ranges (IQR) of 54.3 and 32.9 $mg\ kg^{-1}$ for the biomass and litter sources, respectively. In contrast, the IQR for the organic and Ah sources both were <2.5 $mg\ kg^{-1}$.

Overall, in terms of the areal density of WEP, the top 10 cm of the Ah horizon was the largest source of WEP (42.5 $mg\ m^{-2}$) followed by the biomass (26.3 $mg\ m^{-2}$), organic layer (14.3 $mg\ m^{-2}$), and the litter (13.7 $mg\ m^{-2}$). It should be noted that these are only approximate estimates for the organic layer and Ah horizon because the values for depth and bulk density measured in 2023 were used in the calculations for all previous years (Figure S4). Nevertheless, the vertical profile of WEP in riparian areas (Figure 2) observed in this study supports the concept that a measure of P in soil alone is likely missing a large proportion of the near-surface P that can be potentially lost during the spring snowmelt (Liu et al., 2019a; b; Cober et al., 2019). The substantial proportion of WEP above the soil surface provides evidence that managing the biomass in riparian areas in autumn may reduce the contribution of P lost directly from this area during spring. Specifically, the harvesting of this biomass results in an export of P which can maintain or enhance the buffering or storage capacity of P derived from upslope sources further improving downstream water quality (Kelly et al., 2007; Hille et al., 2019).

Figure 2: Vertical and longitudinal profiles of a) WEP concentration and b) WEP content in the riparian areas prior to grazing and mowing treatments.

The manuscript



Riparian vegetation WEP in response to grazing

Table of contents

- Load Libraries
- Read in data
- Merge data
- Biomass data
- Biomass Analysis
- Dry Weight
- Plots

```
In [17]:  ggplot(data = filter(plot_data, measure == "dryweight"), aes(x =
          treatment, y = diff, fill = location)) +
            geom_boxplot() +
            labs(y = "kg/m^2") +
            geom_hline(yintercept = 0) +
            facet_grid(~measure)
```

A notebook associated with the manuscript

Resources

Online References

- RStudio > Help > **Markdown** Quick Reference
- **Quarto**
  - Quarto Documentation
  - Openscapes' Quarto Tutorial
  - Posit's Welcome to Quarto Workshop! (video)
  - We don't talk about Quarto (blog post)
  - A Quarto tip a day (blog)
- **R Markdown**
  - R Markdown Documentation
  - R Markdown: The Definitive Guide (online book)
  - RStudio > Help > Cheat Sheets > R Markdown Cheat Sheet
  - RStudio > Help > Cheat Sheets > R Markdown Reference Guide

Examples

- This presentation
- Quarto Gallery
- R Markdown Gallery

Resources

Online References

- RStudio > Help > **Markdown** Quick Reference
- **Quarto**
  - Quarto Documentation
  - Openscapes' Quarto Tutorial
  - Posit's Welcome to Quarto Workshop! (video)
  - We don't talk about Quarto (blog post)
  - A Quarto tip a day (blog)
- **R Markdown**
  - R Markdown Documentation
  - R Markdown: The Definitive Guide (online book)
  - RStudio > Help > Cheat Sheets > R Markdown Cheat Sheet
  - RStudio > Help > Cheat Sheets > R Markdown Reference Guide

Examples

- This presentation
- Quarto Gallery
- R Markdown Gallery

Thank you!

Dr. Steffi LaZerte
Analysis and Data Tools for Science

🌐 steffilazerte.ca

✉ sel@steffilazerte.ca