**Week 4 Discussion**

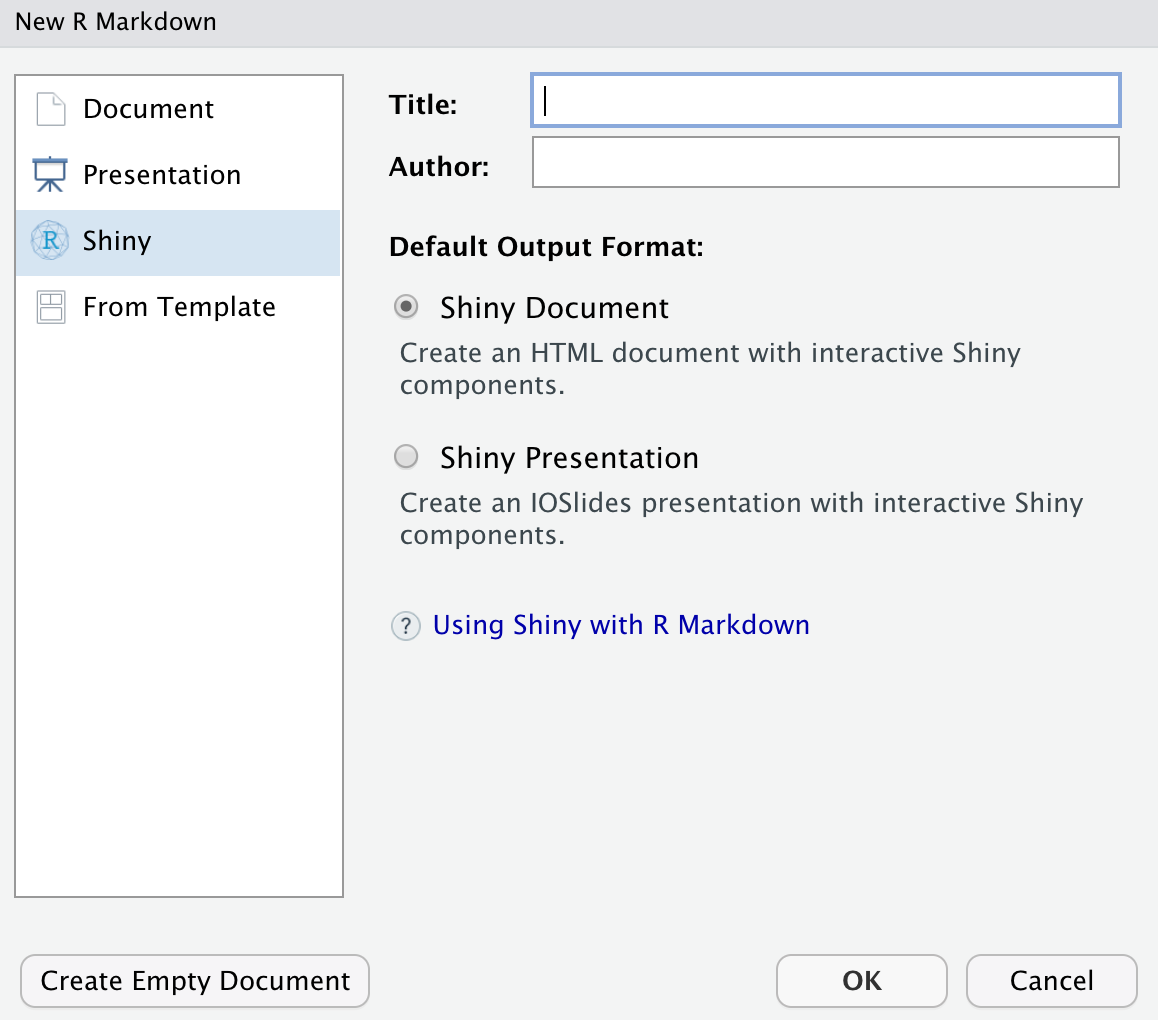
Shiny elements in RMarkdown, publishing an app at shinyapps.io, and some other fun stuff in R

**Part 1: RMarkdown with Shiny elements**

A cool, lightweight option to add interactivity & user choice in RMarkdown outputs

**Step 1.** Create a new GitHub repo (with a ReadMe), clone to connect to a version-controlled R project

**Step 2.** Add a new RMarkdown document, but instead of just a regular RMarkdown document, choose an output of Shiny Document:



Now you have an RMarkdown document that can have Shiny elements embedded within it!

* Run the default document (notice that you have to Run Document, instead of knitting)
* Delete everything below the first code chunk (we’ll make our own widgets & data viz, using some time series skills covered in the ESM 244 recorded lab this week)

**Step 3. Attach packages**

* Attach the **tidyverse**, **here**, **lubridate**, **janitor**, **tsibble**, and **feasts** packages in the setup chunk (install if needed)

**Step 4. Read in the Truckee River time series data (discharge, cfs)**

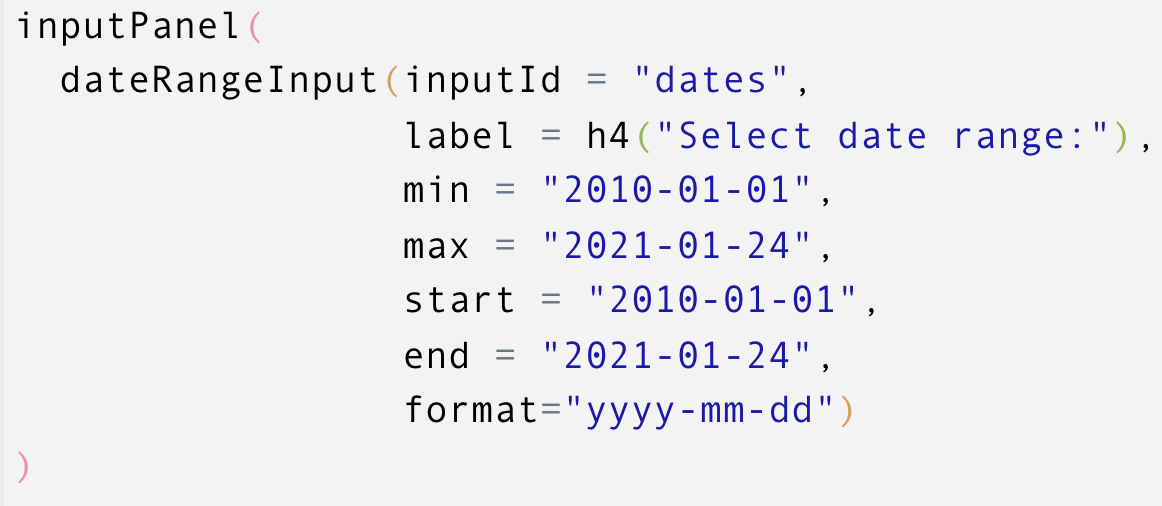
* Download the truckee\_river.csv file from GauchoSpace
* Drop into a data/ subfolder that you add to your project
* Read in the data using here() as object **truckee**, skipping excess rows and simplifying to only keep the **datetime** and **discharge\_cfs** columns

**Step 5. Convert to a tsibble & visualize**

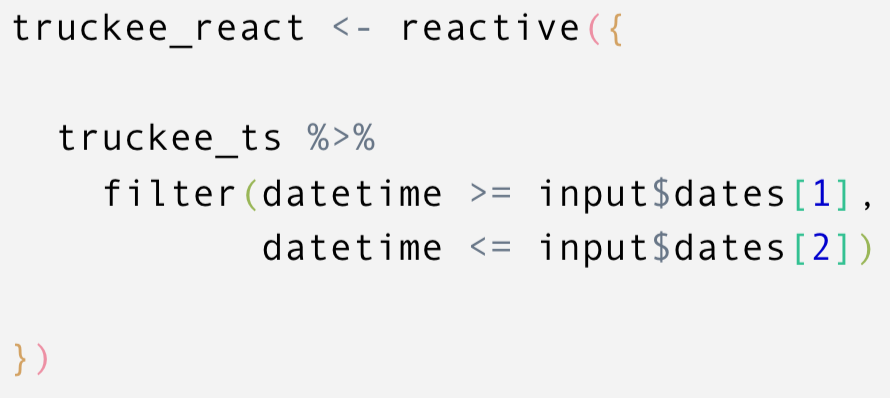
* Convert the date to a date, and the discharge column to numeric
* Convert the data frame to a tsibble
* Create static time series and seasonplot graphs

**Step 6. Embed Shiny elements to allow the user to select a date range**

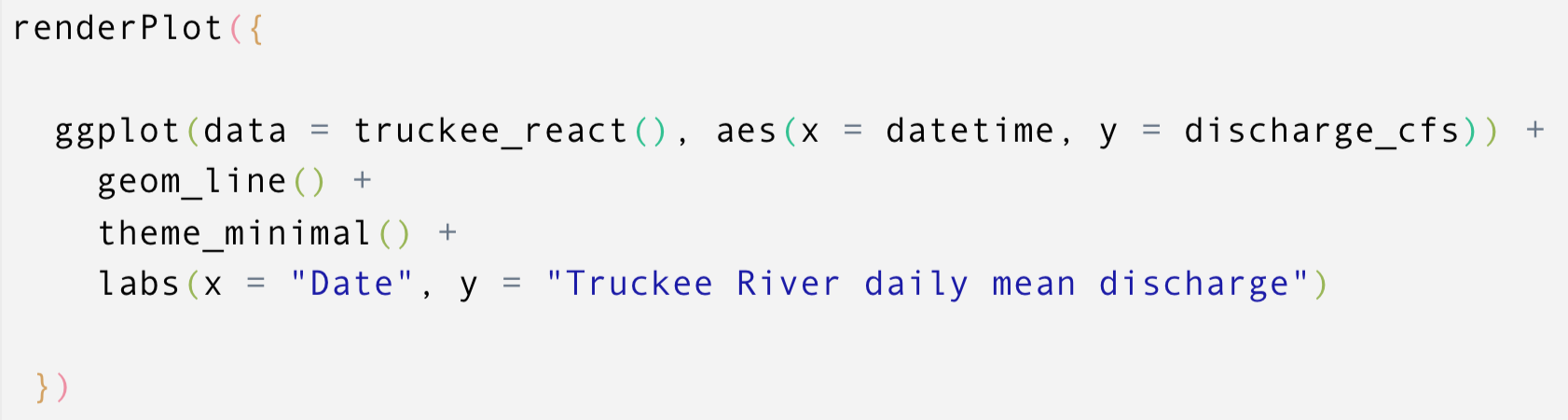
* Create a new code chunk to contain your Shiny element
* Add a widget (dateRangeInput) allowing users to select a range of values:



* Create a reactive data frame using the output from those selections (which is a vector containing two elements, so we use indexing to identify each in our filter step):



* Create a reactive plot that only includes observations within the date range:



**Step 7: Publish (Allison will demo, you’ll need to set up using rsconnect):**

<https://docs.rstudio.com/shinyapps.io/getting-started.html>

**Part 2: RMarkdown tips & tricks**

* Create a new (normal) RMarkdown document in your project
* Save in the root as rmd\_tips.Rmd
* Delete everything below the first code chunk

**Step 1. Populate your document with random headers**

For example, follow along with Allison & Casey to make a template with “Chapter sections” and subsections. Like:

## Introduction

## Background

### R History

### R Today

## My favorite R packages

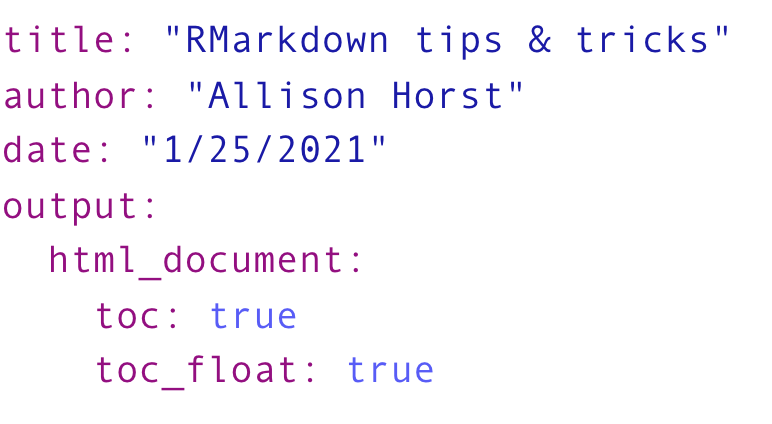
### For data visualization

### For communication

### For analysis

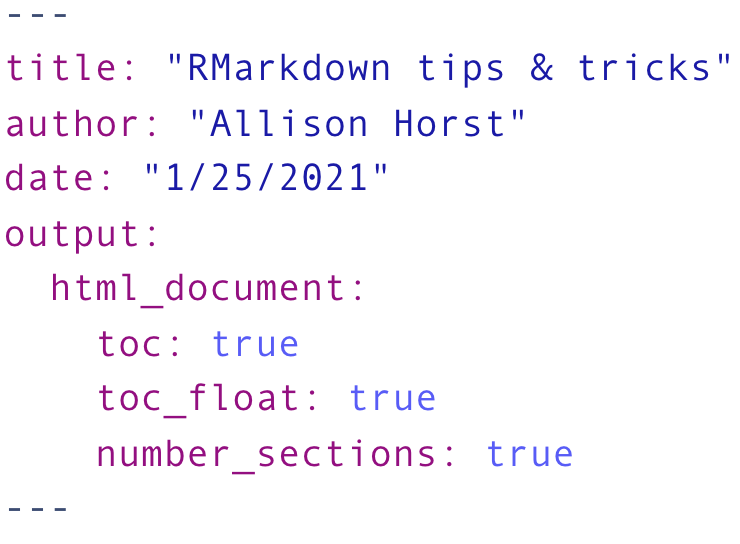
**Step 2. Update your YAML to contain the toc lines below**

Note: YAML is space sensitive)



**Step 3: Want to number the sections? Cool.**

Note that it assumes a level 1 header should be the ones place (so you can try updating some of your headers to level 1 & see how numbering changes).

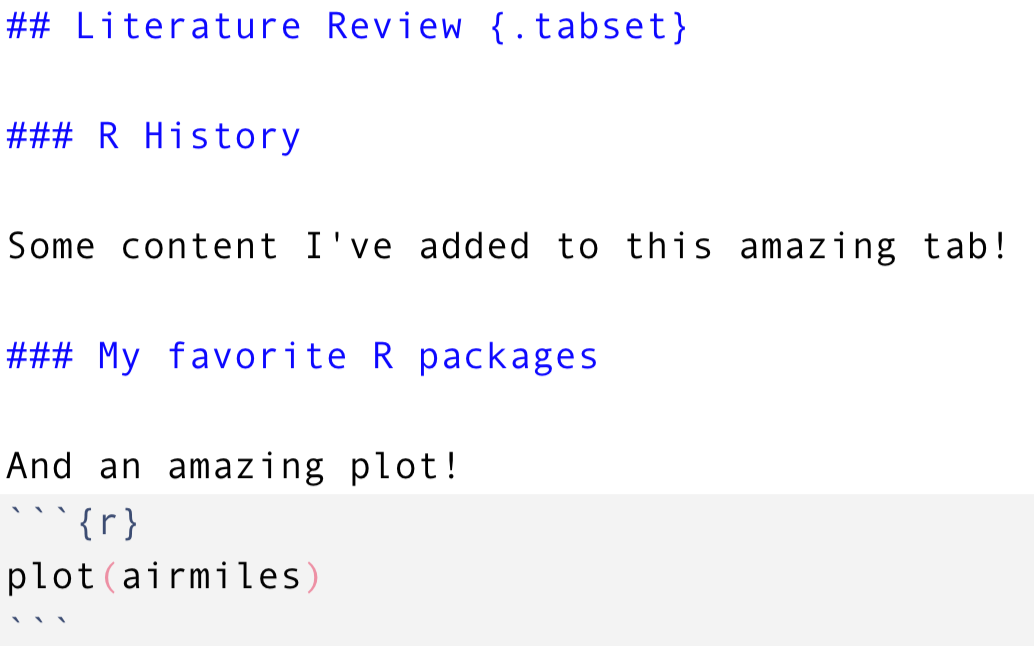
****

**Step 4: Add some tabs within RMarkdown**

Note the parent & child levels

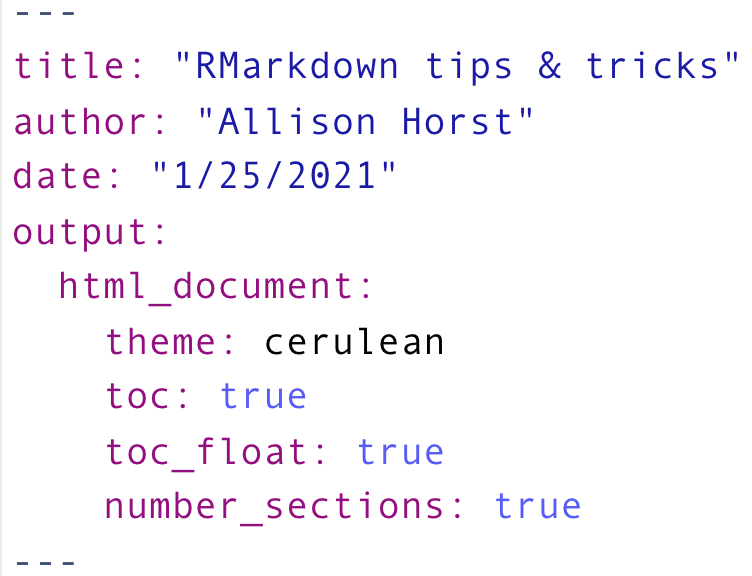
After a header level where you want tabs to exist, add {.tabset}

For example, this will make separate tabs for R History and My favorite R package:



**Step 5: Add an existing bootstrap theme**

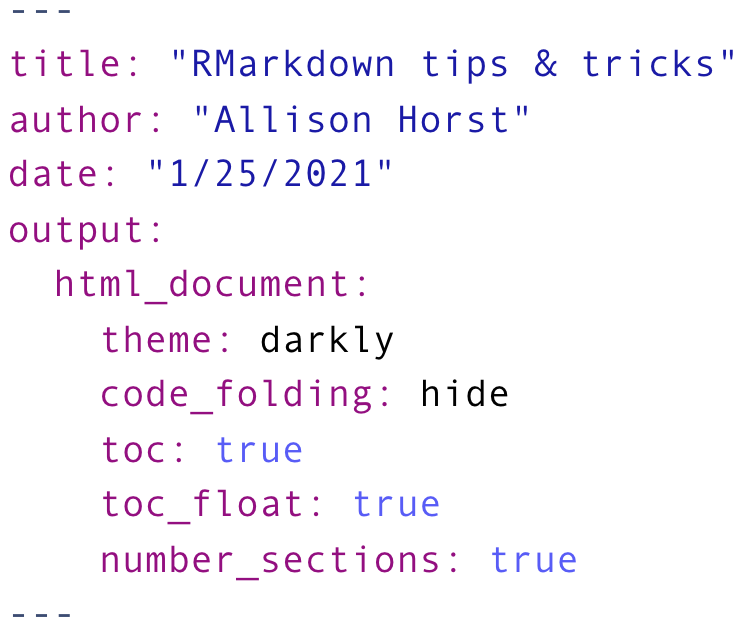
See [HERE](https://bootswatch.com/3/) for bootstrap free theme names. Add the theme name in the YAML, for example:



**Step 6: Code folding!**

Want to have the *option* for someone to view your code but not necessarily have it show up by default? Use code folding!

Add code\_folding: hide to the YAML:



**END!**