Using flipbooks to build outcome-oriented educational tasks

For *Chance Magazine's* 'Taking a chance in the classroom'

Evangeline Reynolds, June 2020

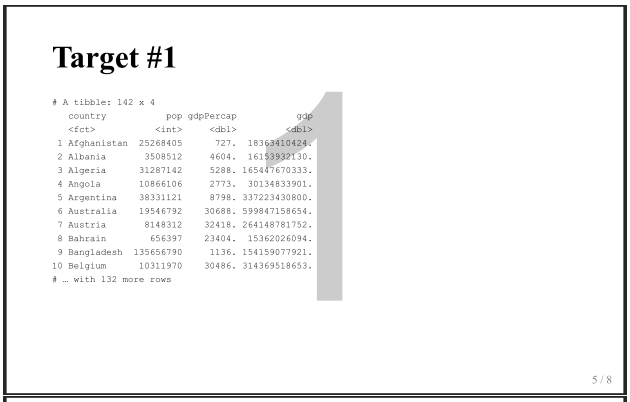
One goal of data science or statistics classes is to give students practice with tools so that they are comfortable preparing and analyzing data. A new way to engage students with such tasks is by using the new {flipbookr} package to build outcome-oriented exercises.

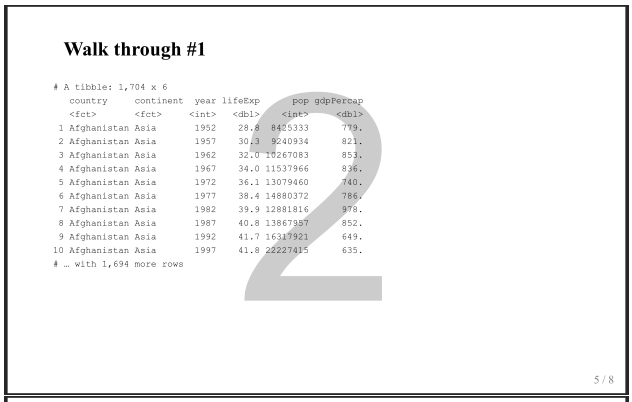
The flipbookr package is a tool that was originally built to show analysis code pipelines (manipulation, visualization, statistical analysis), step-by-step with output in a slide-show presentation. This gives you a movie-like experience linking new code to the change that it triggers in output.

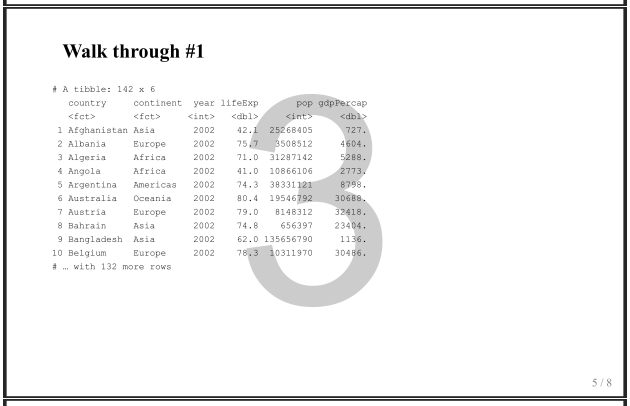
But flipbookr can also be used in a mode where *only output* is displayed. This mode is useful for data science and statistics classroom use! Flipbookr can help educators build outcome oriented tasks.

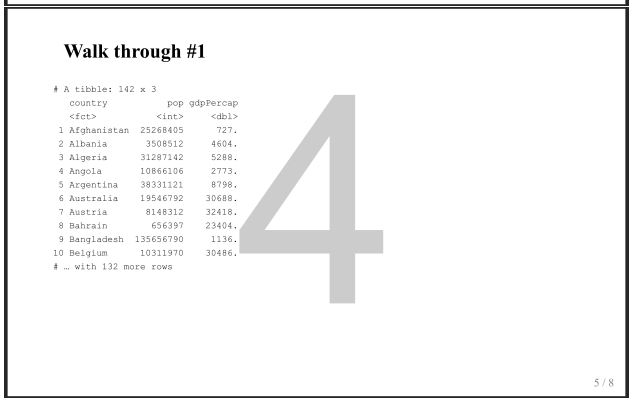
What's this new style of exercise?

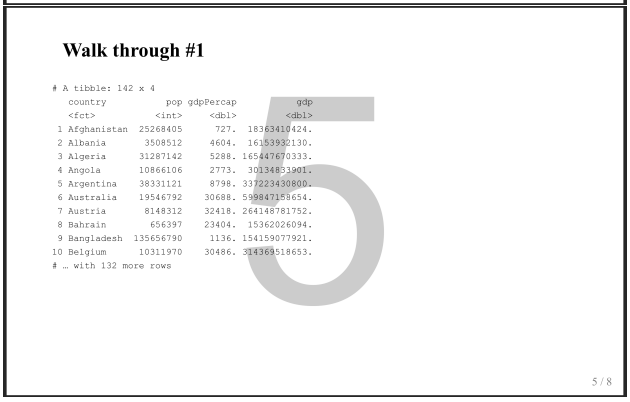
Let's think about an exercise to build up data manipulation practice using flipbookr. First, there might be a reference section of what functions can be used to accomplish the required tasks, or the instructor can present these functions. Then in a flipbook a target is introduced. The target will be a final output of a data manipulation pipeline, where the input is a familiar data set. (I use the 'gapminder' data set. My students know the data set having seen the Hans Rosling presentation, and having worked with it a bit for data visualization.) After presenting the final target, a walk-through from input to target — including intermediary states — are displayed. Students are instructed to 'wrangle to it'; i.e. use data manipulation functions to wrangle their way to the target output. Subsequent exercises (which may become progressively more challenging given more wrangling steps) are structured in the same way; first the target output is presented, and then a walk-though with intermediate outputs. An example of a flipbook containing data manipulation exercise can be viewed here.











What might the student/instructor experience look like?

Some students will be able to determine the steps needed to get to that target, just by looking at the final target. In that case, students can build the data manipulation pipeline without looking ahead. After reaching the target they can then compare their strategy to that of the walk-through.

Other students may not be able to come up with the entire sequence of steps needed to wrangle to the target output. If a student is initially stumped about how to arrive at the final target, they can look ahead to the walk-through that includes the intermediary outputs. These students may be able to accomplish the task by taking it step-by-step. They may be able to at least manage some progress towards the overall target, perhaps getting some assistance from a peer or the instructor if they find a step in the pipeline to be a puzzle.

There's a subliminal 'take-away' message is about problem solving in data science here too: *you may have some big task ahead of you, but stay cool. Try to break up the task into pieces — it will probably feel more manageable.*

As an instructor the task of assisting students who might have trouble may also be more manageable. Students are less likely to be stumped by the entire data manipulation challenge; and even if they are, the task is already logically broken up for the instructor to talk through with students. Students will likely approach the instructor about the specific step in manipulation pipeline rather than an amorphous and possibly exasperated 'I don't know how to get this done!'

Data manipulation as a general problem

I think that an advantage of this outcome-oriented exercise, too, is that it de-emphasizes specific tools. Sometimes the focus of data manipulation exercises is syntactical focusing on the language being used. But this exercise puts the data transformation center stage (subsetting, keeping columns, summarizing etc). In fact, a large number of statistical software or syntactical paradigms can be used to accomplish the manipulation tasks presented in the assignment.

I personally teach and built this exercise using 'the tidyverse' which does have some signatures in the output (like a separate group_by step when compared to data.table for example). But the manipulation tasks could also be solved in Stata, SQL, SPSS, SAS, R tidyverse, base R, data.table, python's pandas – or something yet to be invented! This shifts the emphasis from a *how* question (specific implementation and syntax) to a more general *what* question. The question "what are the data manipulation moves that I can make?" is primary, and how to do that is secondary.

The example that I link to has about 15 wrangling pipelines — a fair amount of practice for beginning to internalize new tools and modes of thinking. The first half I call 'one-stream' data manipulation and the second half is on summarizing data. Educators should feel free to use this exercise as is or can create their own outcome-oriented tasks using the new {flipbookr} package!

How to build your own exercise

If you are familiar with R and Rmarkdown, you're in a good place to start building these outcome-oriented educational tasks. Still some things will be new and I'll discuss them here, but it will also be useful to look at a basic tutorial on creating your first flipbooks and a short example of how to build an outcome-oriented exercise that I'm proposing toy exercise.

First, to get the movie-like experience of flipbooks, we need a slide show tool. The flipbookr package is used with the Rmarkdown tool xaringan to deliver material to .html slides. Your rmarkdown file will need to have the appropriate meta data set up so that these slides are produced.

Then you can write a code chunk with steps of a data manipulation task. The code chunk options should be set to echo = FALSE and eval = TRUE so that the result displays but the code is hidden. After a slide break, you'll then use the flipbookr function chunk_reveal to refer to the task chunk. You should set `display_type` to "output" so that only the output is shown, but the code to accomplish the steps is hidden.

````
---

# Target #1

```{r task1, eval = FALSE}
cars %>%
  ggplot() +
  aes(x = speed) +
  aes(y = dist) + #BREAK
  geom_point(
             shape = 21,
````
````

```
              size = 7, #BREAK2
              color = "magenta", #BREAK3
              alpha = .8 #BREAK4
              ) + #BREAK
  aes(fill = speed) +
  scale_fill_viridis_c() +
  theme_minimal() #BREAK
```

`r chunk_reveal(chunk_name = "task1", display_type = "output", title = "## Walk-through #1" )`