

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 and statistics courses 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 an *only output* mode. This mode might be useful for data science and statistics classrooms!

What's this new style of exercise?

A data manipulation exercise built with *flipbookr* might look as follows. First, you might have a reference section of functions that can be used to accomplish the required tasks. Then a target data frame is introduced. The target will be a final output of a data manipulation pipeline, where the input is a familiar data set. For example, 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-through with intermediate outputs is shown. An example of a collection of slides that would constitute one of the wrangling exercise follows and a full set of wrangling tasks for an assignment can be viewed here (<https://tinyurl.com/y6b9z5vx>).

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 advancing step-by-step. They may be able to 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.

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!’

There’s a subliminal take-away message 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.*

Data manipulation as a general problem

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

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 the how-to syntax 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. Educators should feel free to use this exercise as is or can create their own outcome-oriented tasks using the new `{flipbookr}` package (<https://github.com/EvaMaeRey/flipbookr>)!

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 briefly. For more detailed guidance it will also be useful to look at a basic tutorial (https://evamaerey.github.io/flipbooks/flipbook_recipes) on creating your first flipbooks and a short example (<https://tinyurl.com/y25d2nhz>) of exactly how to build an outcome-oriented exercise.

First, to get the movie-like experience of flipbooks, we need a slide show tool. The `flipbookr` package is designed to be used with the Rmarkdown package `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.

Once you have this in place, you can then 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 output displays but the code is hidden. After a slide break, you’ll then use the `flipbookr` function `chunk_reveal` to refer to the prepared chunk. You should set `display_type` to “output” so that the output is shown but the code used to accomplish the manipulation is hidden.

```
---  
  
# Target #1  
  
```{r task1, eval = FALSE}  
mtcars %>%
 select(mpg:wt) %>%
 arrange(-mpg) %>%
 slice(1:5)
```
```

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

You'll probably want to host the resulting html and associated files online for students. Github Pages has worked well in my experience.

Going further

A bit more may be asked of students for further engagement with the pipelines they have built. At the end of data manipulation pipelines, you might ask students to create a descriptive name for an object that will hold the manipulated data or describe the data manipulation steps in words in groups. Also, educators might think about creating outcome oriented tasks like these for data visualization or data analysis!