Student grades case study

STA303/1002 Week 2

2021-01-18

Getting Started

This is an R Markdown Notebook. When you execute code within the notebook, the results appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Cmd+Shift+Enter* (Mac) or *Ctrl+Shift+Enter*.

```
print("Hello world!")
```

```
## [1] "Hello world!"
```

```
x <- 2 + 2
x
```

```
## [1] 4
```

Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Cmd+Option+I*.

When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Cmd+Shift+K* to preview the HTML file).

The preview shows you a rendered HTML copy of the contents of the editor. Consequently, unlike *Knit*, *Preview* does not run any R code chunks. Instead, the output of the chunk when it was last run in the editor is displayed.

Markdown and R Markdown

Markdown is a document processor, newer and simpler than LaTeX. It is a lightweight markup language with plain text formatting syntax (i.e. you don't have to click lots of formatting options like in Word, you just write symbols). A potential issue is that it can be hard to debug errors in your .md file.

R Markdown lets you integrate R code and Markdown to create reports with commentary, code, outputs and graphics all in one place with no need to copy and paste things in to a Word doc.

Making a .pdf

When you press *Knit* a .pdf report is written. When you do assignments that is what you will want to submit along with your .Rmd file.

A few R Markdown tips

1. There is a small button at the top left of your source area of R Studio that when pressed will show an outline of the document, based on headings in the file. A heading has the hash sign (you might call it a number sign) in front of it. One hash sign is the biggest heading, followed by two hash, and three hash.
2. A range of keyboard shortcuts can be found here.
3. On the top menu bar, go to Help > Cheatsheets > R Markdown Cheat Sheet to open the main R Markdown cheatsheet.
4. On the top menu bar, go to Help > Markdown Quick Reference to open the *Markdown Quick Reference* in your help pane.

Note: You can use `\newpage` in R Markdown to create a page break, which will be helpful when you're preparing assessments for Crowdmark and want questions to be on separate pages.

Case study

The premise

Suppose you're in grad school at your dream univeristy and are the Head TA for a course called STA101. At the end of semester, the instructor gets sick and needs to have major surgery and so will not be able to do the final admin for the course. The department Chair asks for your help in preparing the final grades. They tell you they'll check over everything for you before the marks are submitted, but you want to make sure you do it right and provide clear and reproducible code so you can impress the Chair.

Thankfully, all the marking is done, and the prof has left you the following files and instructions.

```
list.files("student_data/")
```

```
## [1] "STA101_2-perc-survey.csv"
## [2] "STA101_assessment-scheme-from-syllabus.RDS"
## [3] "STA101_course-roster.csv"
## [4] "STA101_gradebook-from-LMS.xlsx"
```

Your prof's message

Hi,

Thanks for helping out with this, I owe you a 100 coffees next semester when I'm back and recovered.

"STA101_gradebook-from-LMS.xlsx" contains the grades from our learning management system. It covers everything except for the 2% survey we did in Microsoft Forms. For the survey, you have "STA101_2-perc-survey.csv", which has the list of emails of the students that completed the survey. "STA101_course-roster.csv" has each student's name, student ID number, email and tutorial group, it should help you join up the data.

You can find it in the syllabus, but also have given you "STA101_assessment-scheme-from-syllabus.RDS" that has how much each assessment (or set of assessments) is weighted for the final mark. It also has how many points the assessment was out of.

I've already dealt with all the accommodations, so an `NA` in the data from the learning management system means the student didn't submit the assessment and didn't get an accommodation, i.e., they get a 0 on that task.

Also, so many students who are close to a GPA threshold will email once final marks are out asking to be bumped up. I'd never want to be unfair and give a student an advantage just because they asked, where another student in the same situation wouldn't get the advantage because they were too polite to ask. So, I was planning to use the below function to adjust everyone who was 1 percentage point below the threshhold up to it. If the Chair asks you to help manage the emails afterwards, tell anyone who asks that the course policy is to make no ad hoc adjustments for individual students as that wouldn't be fair.

```
adjust_GPA_thresh <- function(x){
  thresh <- c(89, 84, 79, 76, 72, 69, 66, 62, 59, 56, 52, 49)
  ifelse(round(x)  %in% thresh, round(x)+1, round(x))
}
```

For submitting the marks, you'll need one file per tutorial group, with just their student id and final grade (rounded to the nearest whole integer). Don't have column titles, it'll mess with the system.

Please also provide the undergraduate administrator a graph of the distribution of the final grades, and if you can, indicate tutorial group too.

Good luck! Your grateful prof

Credits

- 'Student' names come from: https://dataverse.harvard.edu/file.xhtml?persistentId=doi:10.7910/DVN/28201/VEG34D&version=1.0
- Simulated grades created using this little bounded rnorm variation: https://stackoverflow.com/questions/19343133/setting-upper-and-lower-limits-in-rnorm