Week 10 Update

51 problems

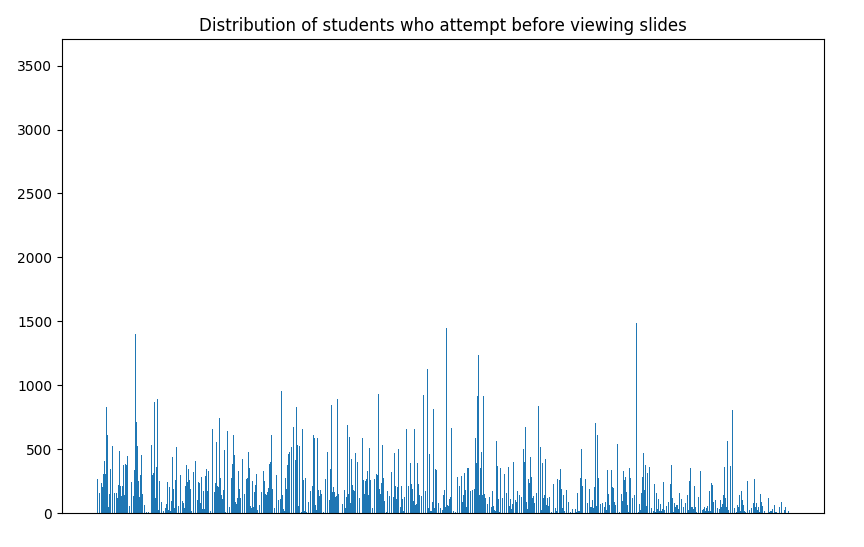
48 problems where it does occur

3 problems where it does not - ['challenge-advanced-2018big-two', 'challenge-advanced-2018intro', 'challenge-advanced-2018w4-python']

14164 students

23499 times this has happened

Students do this on average twice



The x-axis represents individual student IDs and the y-axis represents the number of times a student has attempted a problem before viewing slides. The figure was created by using a dictionary that stored student IDs as keys, and the number of times that they attempted a problem before viewing the prerequisite slides. From the figure, we can see that most students having counts of less than 500 and a minority of students (to do: count later) having counts of over 1000. We might need baseline – look at the proportion – calculating the average, STD – as a proportion of all the slides they access. The students might have viewed the slides before in a previous week. It is a reasonable approach to say that for that particular module, whether they access the slides before they access the problems because the way the content is arranged. There are some dependent issues – knowing how to get the length of the string might be knowledge from several weeks ago. The more immediate stuff – new content.

This is a reasonable filter to say that for the content for that module.

Conclusions: Need to present data slightly differently – aside from normalisation. Hard to interpret the data. Sort the data according to the y-axis. You get more of a histogram – PDF/CDF. Not sure what data to make about the data at the minute.

Statement that we would like to articulate: X proportion of students access the problem before the prerequisite slides.

10 modules, 2 problems per module. There should be about 20 problems. There needs to be a bit more filtering set going through. Double counting or multiple counting they visit the same problem slide.

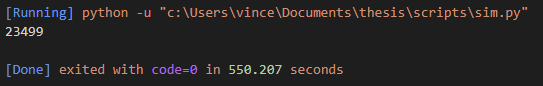
There is something that is incorrect in the code.

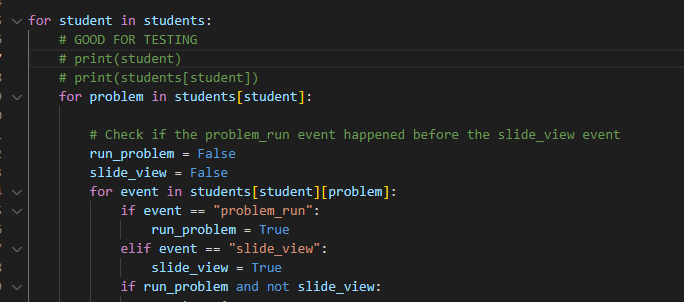
300,000 2

I

THINGS TO DO

* Check the code again
  + Ensure that you are not double counting
  + Produce new results with the distribution
  + Try to get code to be quicker
  + PRODUCE THE CDF
  + Get the code even quicker – search iterruplse alternatives
* Filter the dataset
  + Come up with the different events and ways to filter them (based on interaction sequences for each user and each problem – i.e. students[current\_student][current\_problem]
    - Attempt problem without viewing the slides at all (problem\_run occurs before slide\_view)
  + Focusing on one problem, filter the data set based on the event types
* Repeat for all the other slides
* Present results as a CDF





f we look at the first problem of the module – first problem that they look at as opposed to the first problem overall

Several combinations of click events:

1. If a student goes through the problem and looks at a slide, might not understand, then they hunt around. You could identify slides that have the most important content.
2. Overall the slides before provide the content.
3. Event (label these events/people who engage)
   1. Visit before doing: Do people visit the problem, and then look at the content.
   2. Engage with material and then visit: Other one is they go through the slide, and go through the problem.
   3. Some slides have a completion event. Do people complete the activity slides before attempting the problem.
   4. We need to be careful about attempt – visit the problem vs running the code and submitting the code.
   5. Attempt problem without viewing at all.
   6. There will be distribution across all different scenarios
4. Do the filtering
   1. Take a look into bugs in the code
   2. If you want to refine things further you can go and look at the problems, and then manually classify slides – annotate the problems. You could narrow down your reporting into – did they do these slides for the problem.
5. Next steps
   1. See what is happening manually – look at the problem itself. To guide your analysis, look at a particular example – if you look at the Week 1 Module 1. Manually go to that problem and look at the slides. What are the most important slides here. What do you think are the important things that they should do to make the subsequent problem easier to important.
   2. Check the code again, then produce new results with the distribution
   3. Filter the dataset according to these types of events
   4. DO THAT FOR JUST ONE PROBLEM FIRST – if that works out then expand it to all the other problems – have more background context to understand the data.
6. Are some slides more useful than others? (identifying usefulness of slides – earmarking for thinking about later).

SLIDES: Not visited at all, visited once, visited in a sequential order, visited in a random order – it might be easier to look for specific patterns (if others don’t follow the pattern, then expand). There may be overconfident students. Students may look at slides out of sequence 0 strategic students but not deep learners – some students spend minimal effort to get the maximum return – not engaging with the material. Show up later when problems become more difficult.

Sequential process: is the one that we would like people to do (identify this).

It’s possible for us to score this approach. Create surface and deep scores (0 to 1). The absolute deep learning one – doing the thing as it was intended/written – start at first slide, complete every slide, move onto the next one and complete all the steps in each slide. That’s the intended way in which the course was written.

There are also people who skip through slides – think they understand it, don’t interact with it – cursory tour of the slides. (person who does all slides, person who tours through slides, person who attempts problem straight away – so many strategies – we should categorise these in some way).

Start off by identifying these specific strategies hypothetically

Look for the counts of those and their distribution

And then branch out

There may be some theory behind this – so that we can corroborate the theory with this. Back this up with theoretical knowledge.

Bryn – lot of interesting stuff in here. It’s not having to do a lot of detailed analysis. It would be nice to go through a simple paper to begin with. It would be great if we could get something out. Get some headway on this analysis – we could write quite an accessible paper just on this kind of thing. We could spend time looking at graphics. One of the reason why people might skip through slides is because they can’t read very well – avoid the dense/text-heavy slides. There might be some demographic differences. People might change approaches over time.

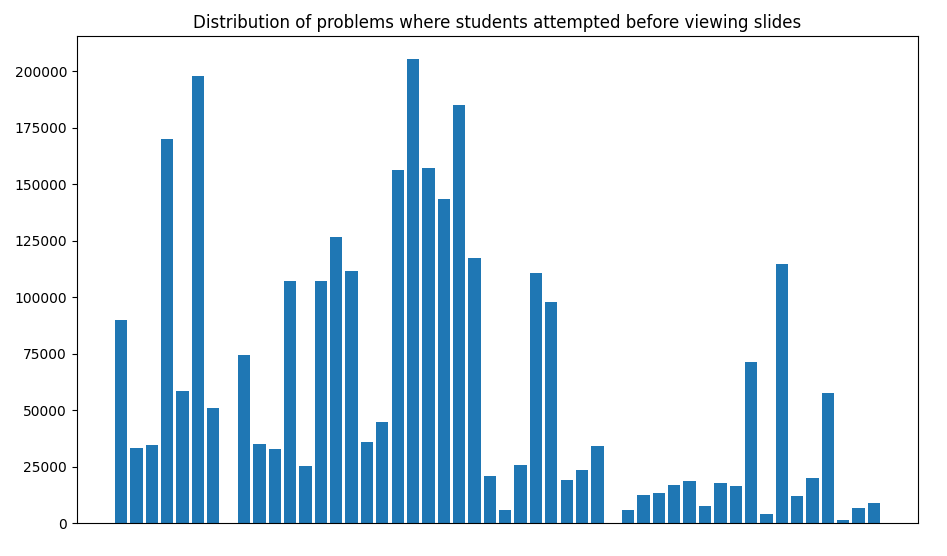
Get some visually pleasing outcomes.

**Call for papers: April 22 – July 2** (TRY TO GET A PAPER BY THEN – short paper).

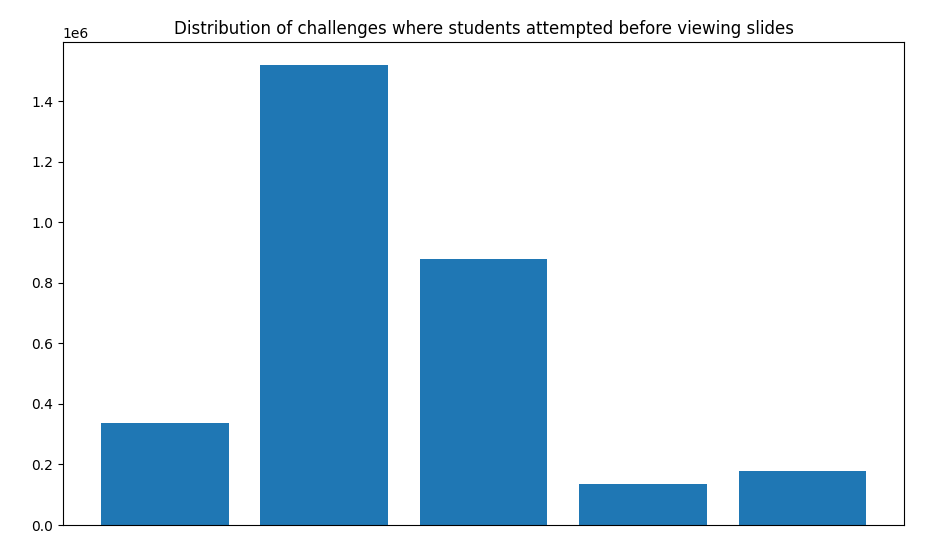
Look through Sophia’s analysis – might help thinking about how that aligns with current processing procedure. May help to ground to just look at one or two earlier modules.

[BY JUNE 23 FREE](https://ascilite.org/annual-conference/) – **get the results before then (keen to help with this paper). Provide the scaffolding for writing a paper the first time around.**

**PUT NUMBERS TO THOSE THINGS – STATS**



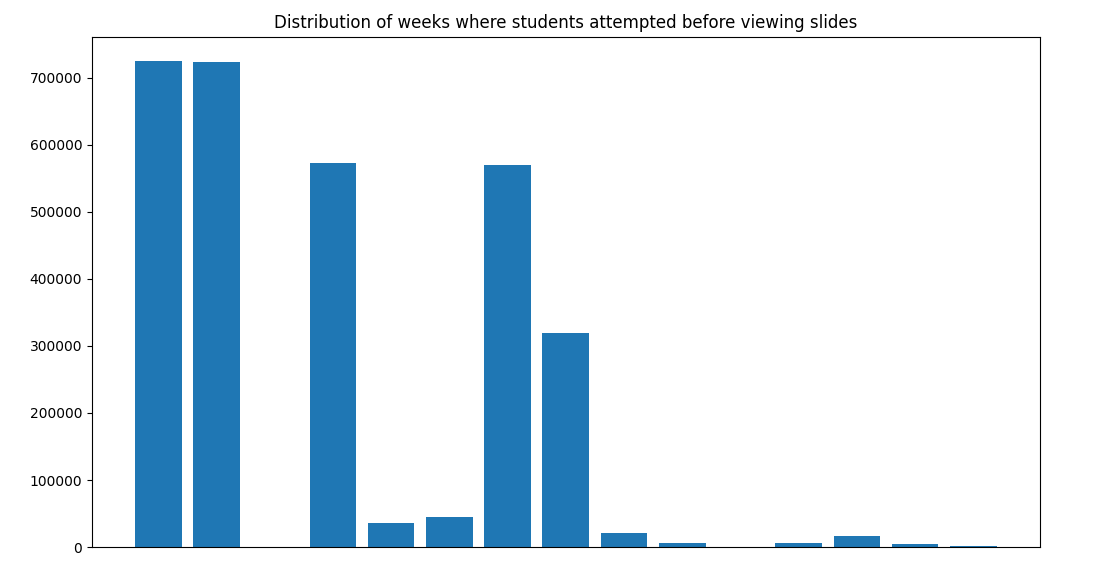
The x-axis represents individual problems and the y-axis represents the number of times a student has attempted a problem before viewing slides. The figure was created by using a dictionary that stored problem names as keys, and the number of times that students had attempted said problem before viewing the prerequisite slides. From the figure, we can see that there are a number of problems where students disproportionately attempt the problem first before viewing the prerequisite slides.



Columns

1. challenge-newbies-2018': 336117
2. 'challenge-beginners-2018': 1518831
3. 'challenge-intermediate-2018': 877588
4. 'challenge-advanced-2018': 136837
5. 'challenge-beginners-blockly-2018': 178127

The figure shows that the challenge-beginners-2018 challenge had the highest rate of students attempting problems before they had viewed the prerequisite slides.



Weeks:

* 'w1': 725055
* 'w2': 723943
* 'w4-': 1
* 'w3': 572891
* 'w1-tournament': 36126
* 'w1-': 44884, 'w4': 569203
* 'w5': 319571
* 'w2-tournament': 20879
* 'w2-': 5976
* 'big-two': 9
* 'w3-': 6071
* 'w3-tournament': 17117
* 'w4-tournament': 4133
* 'w5-tournament': 1641

The figure shows that the challenge-beginners-2018 challenge had the highest rate of students attempting problems before they had viewed the prerequisite slides.