Welcome Everyone to the

From Traditional Lectures to Flipped Classroom and Automated Assessment: how did we do it?

Maurício Aniche and Frank Mulder

This is a **hybrid** session and will be **recorded**.

Please turn on your camera and mute your microphone.

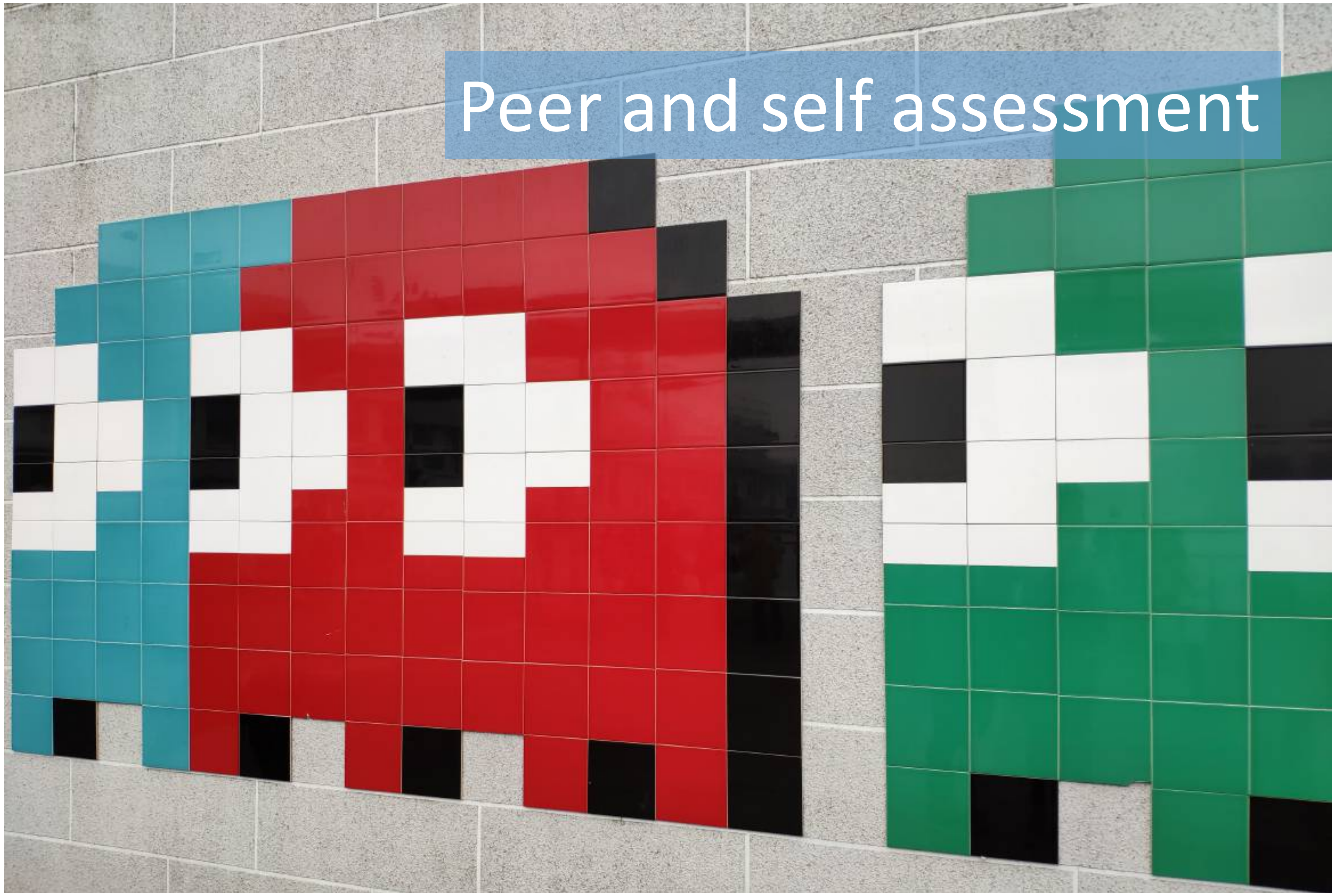
Questions/remarks are very welcome:

- During the presentation, please use the chat for questions/remarks.
- After the presentation, feel free to unmute your mic (or use the chat).

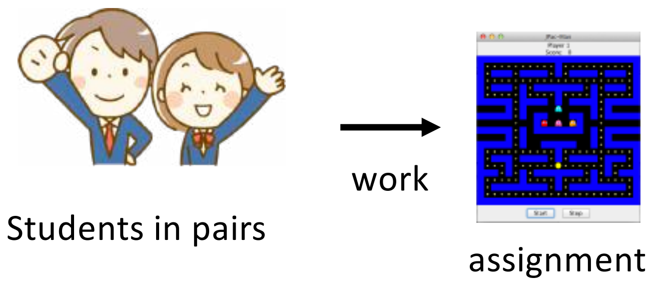
TUDelft | Teaching Academy

THE EXPERIMENT | 4 NOV'21

Back in 2016

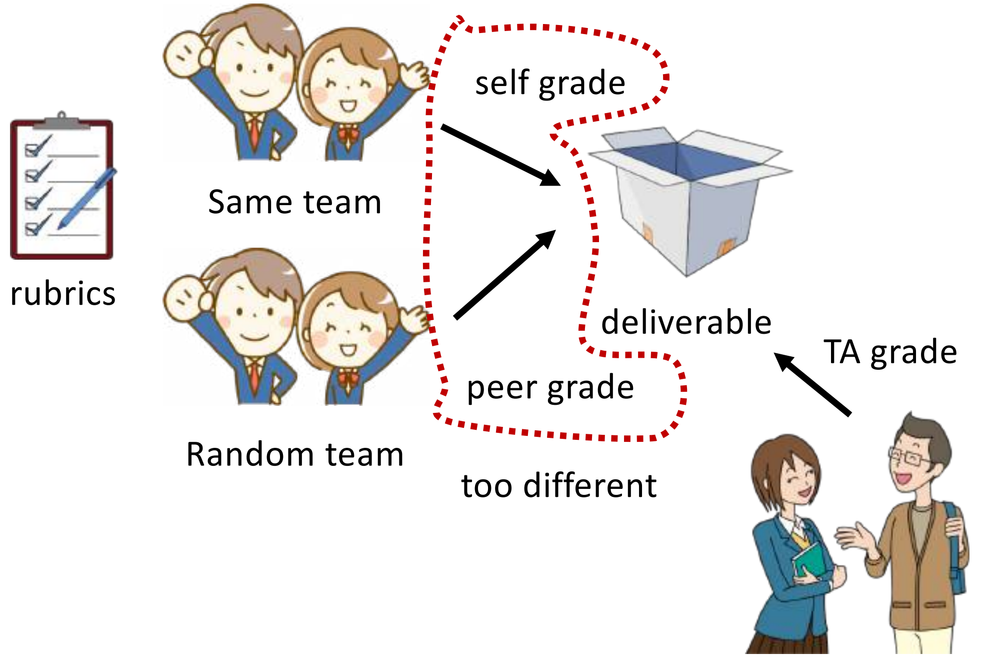


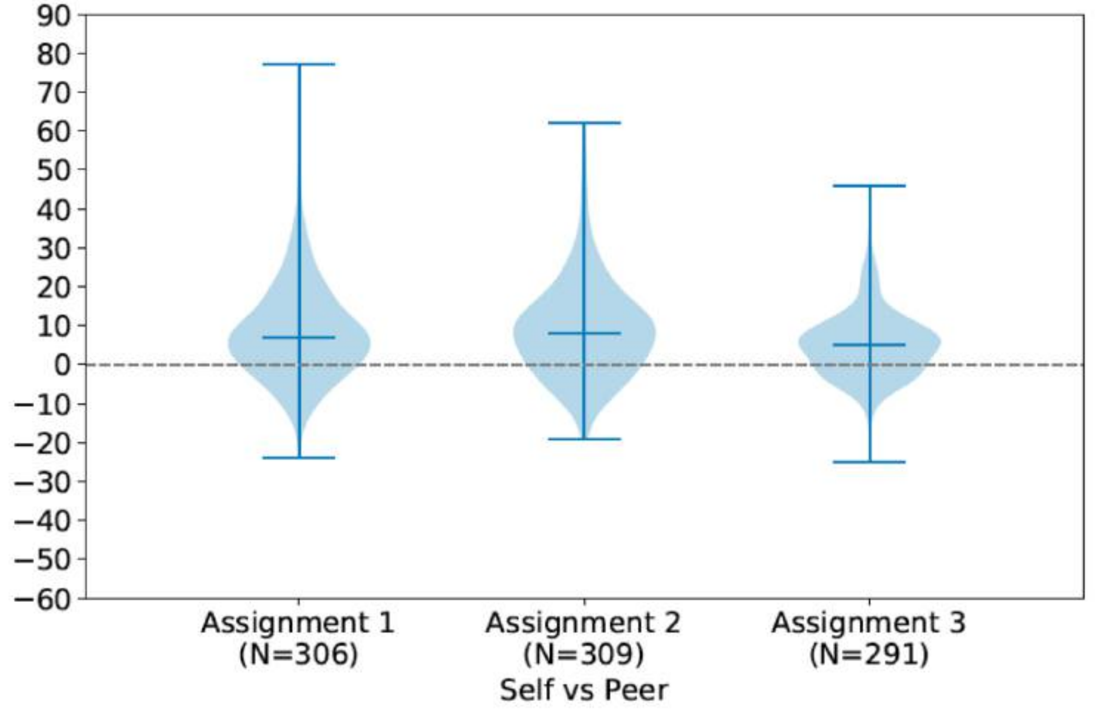
Peer and self assessment



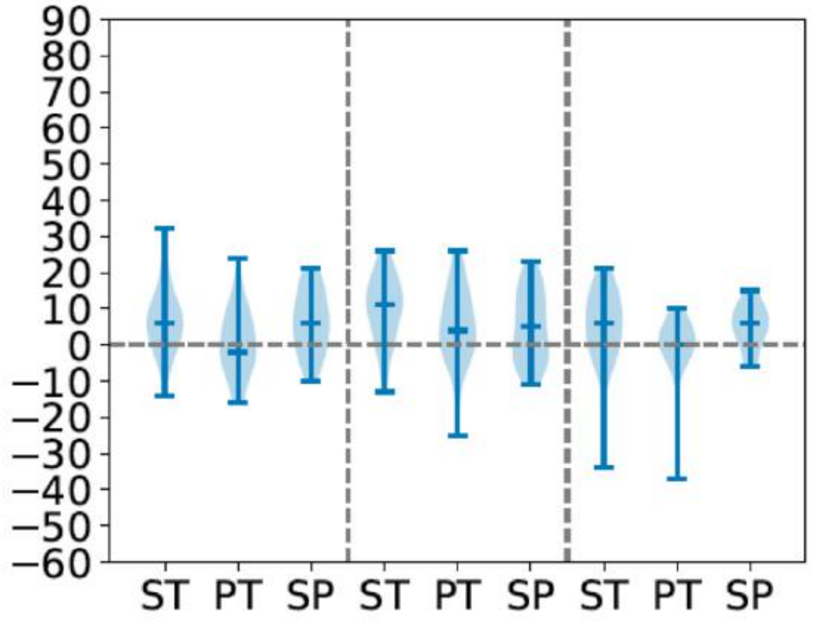
906 self and peer graded submissions

248 of them double checked by our TAs



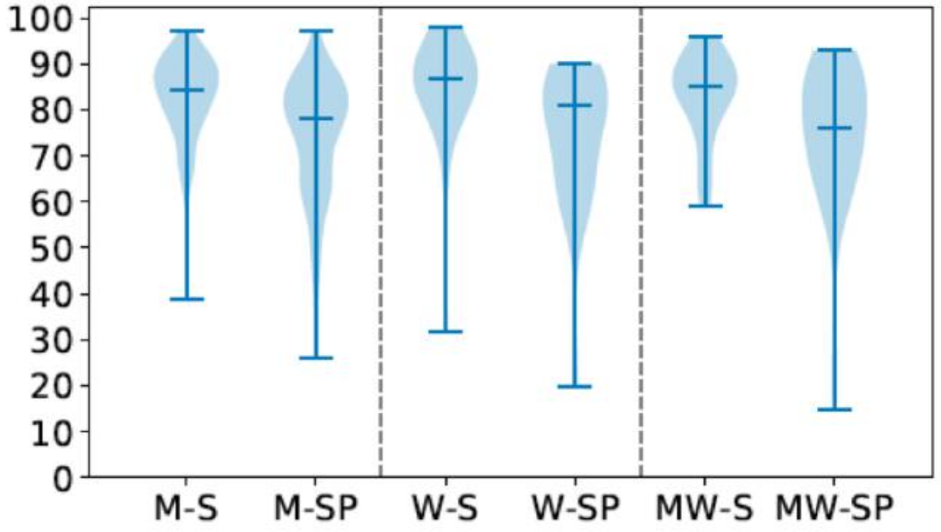


- Self grades tend to be 8–10% higher than peer grades.
- Around 25% of the teams give themselves a self grade lower than their peers.
- Precise matches between the self and peer grade rarely happen.



(c) Stratum 3: Random teams. 1st assignment N=29, 2nd assignment N=43, 3rd assignment N=27.

- Peer grades seem to be a good approximator of TA grades.
- In cases where self and peer grade diverge significantly, the TA grade appears to lie in between.

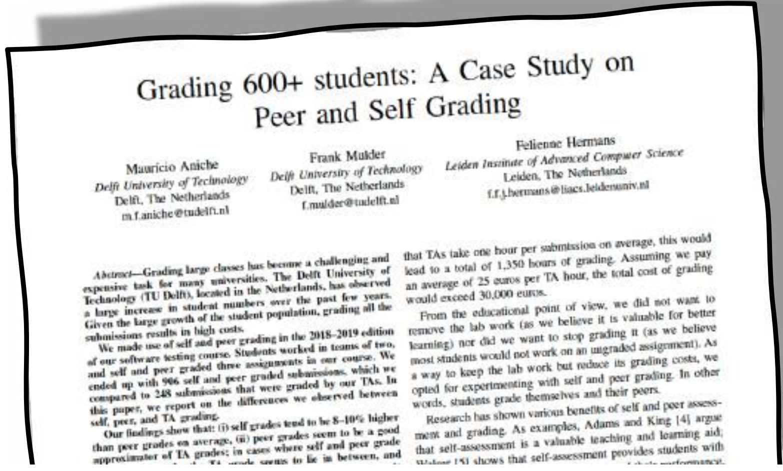


(a) Assignment 1

Gender and nationality do not seem to affect the way teams perform self and peer grading.

Takeaways

- Self and peer grades seem a viable alternative for grading lab assignments.
- In practice, we went for the higher grade between self and peer grade, or for the TA grade when we had it.
- Careful with the extra workload for the students.
- The importance of the rubrics.
- Still careful with gender biases.



Grading 600+ students: A Case Study on Peer and Self Grading

Mauricio Aniche
Delft University of Technology
Delft, The Netherlands
m.f.aniche@tudelft.nl

Frank Mulder
Delft University of Technology
Delft, The Netherlands
f.mulder@tudelft.nl

Felienne Hermans
Leiden Institute of Advanced Computer Science
Leiden, The Netherlands
f.f.j.hermans@liacs.leidenuniv.nl

Abstract—Grading large classes has become a challenging and
expensive task for many universities. The Delft University of
Technology (TU Delft), located in the Netherlands, has observed
a large increase in student numbers over the past few years.
Given the large growth of the student population, grading all the
submissions results in high costs.
We made use of self and peer grading in the 2018–2019 edition
of our software testing course. Students worked in teams of two,
and self and peer graded three assignments in our course. We
ended up with 906 self and peer graded submissions, which we
compared to 248 submissions that were graded by our TAs. In
this paper, we report on the differences we observed between
self, peer, and TA grading.
Our findings show that: (i) self grades tend to be 8–10% higher
than peer grades on average, (ii) peer grades seem to be a good
approximator of TA grades; in cases where self and peer grade

that TAs take one hour per submission on average, this would
lead to a total of 1,350 hours of grading. Assuming we pay
an average of 25 euros per TA hour, the total cost of grading
would exceed 30,000 euros.
From the educational point of view, we did not want to
remove the lab work (as we believe it is valuable for better
learning) nor did we want to stop grading it (as we believe
most students would not work on an ungraded assignment). As
a way to keep the lab work but reduce its grading costs, we
opted for experimenting with self and peer grading. In other
words, students grade themselves and their peers.
Research has shown various benefits of self and peer assess-
ment and grading. As examples, Adams and King [4] argue
that self-assessment is a valuable teaching and learning aid;

Flipped classroom

Type to search

Software Testing: From Theory to Practice

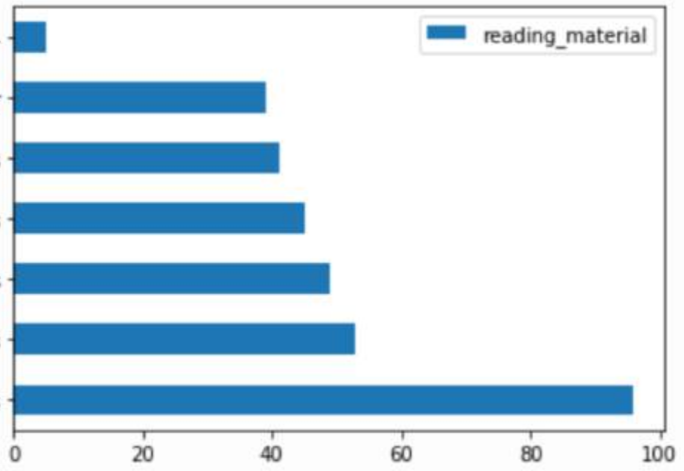
Welcome to **Software Testing: From Theory to Practice**!

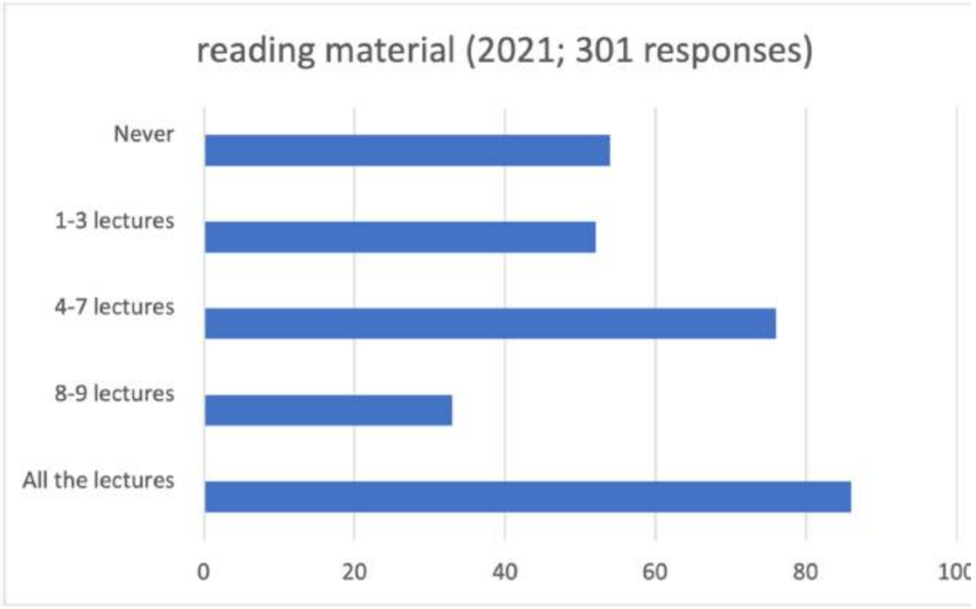
This book covers the most important testing techniques needed to build high-quality software systems. Specific topics covered are quality attributes, maintainability and testability, manual and exploratory testing, automated testing, DevOps, test adequacy, model-based testing, state-based testing, decision tables, reviews and inspections, design-by-contract, test-driven design, unit versus integration testing, mocks and stubs.

We expect readers to be able to:

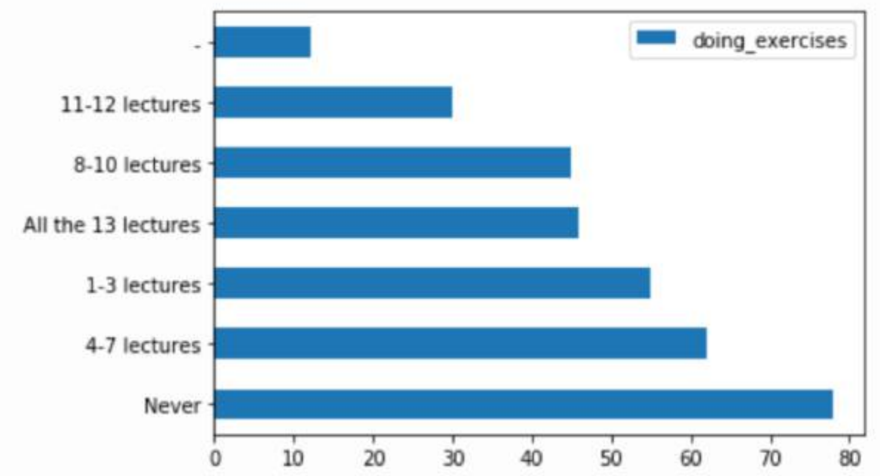
- Create unit, integration, and system tests using current existing tools (i.e., JUnit, Mockito, and JaCoCo) that effectively test complex software systems.

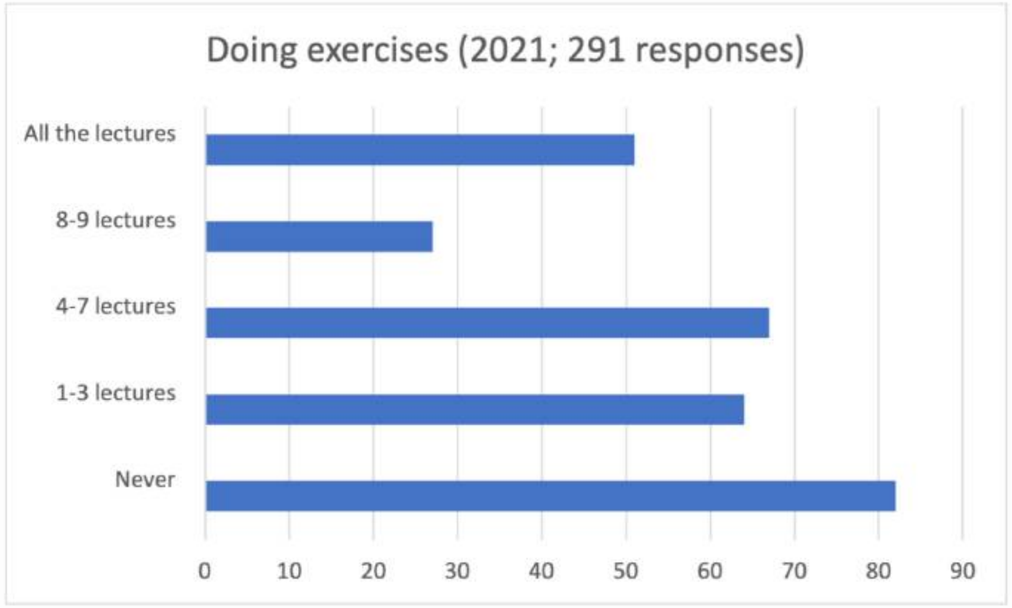
Students read the lecture notes!



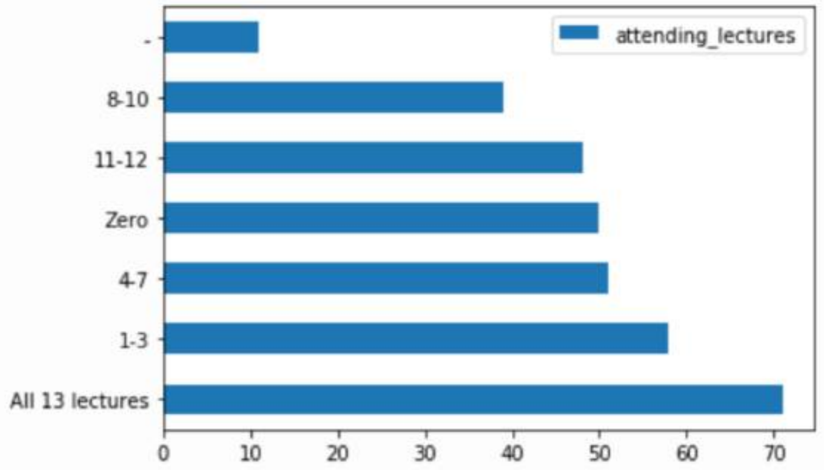


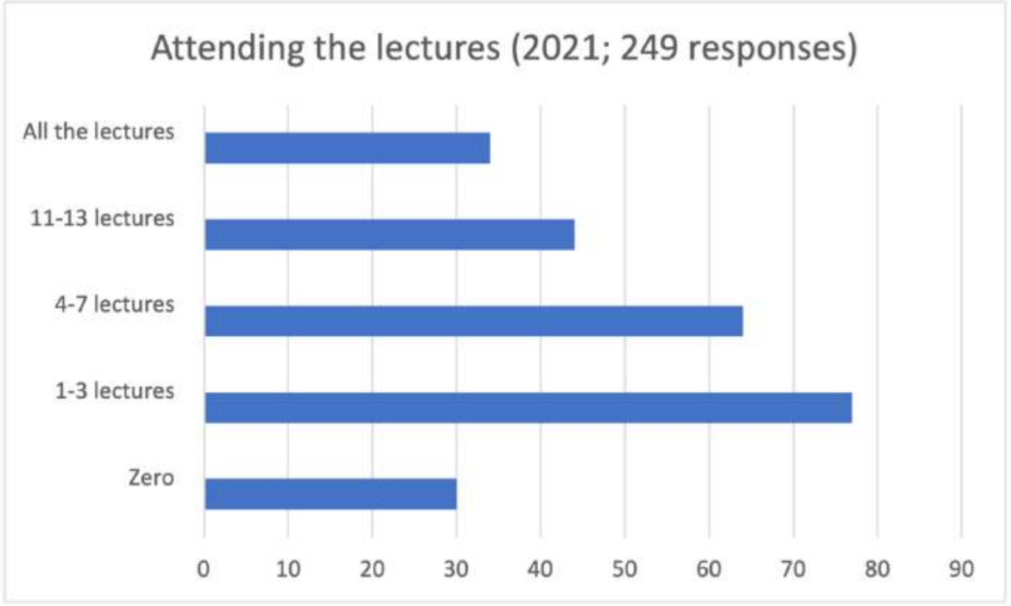
Doing the exercises before the lecture does not happen…



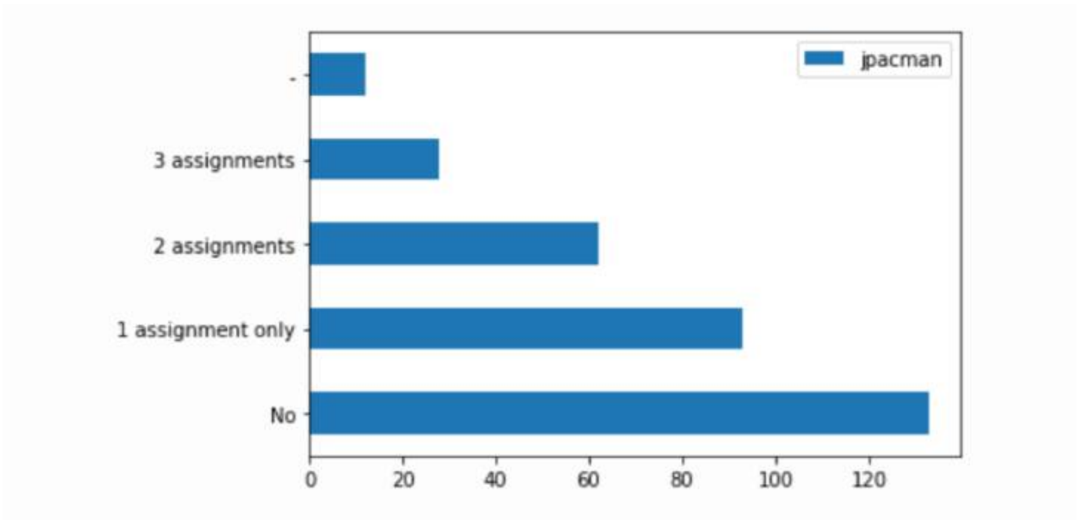


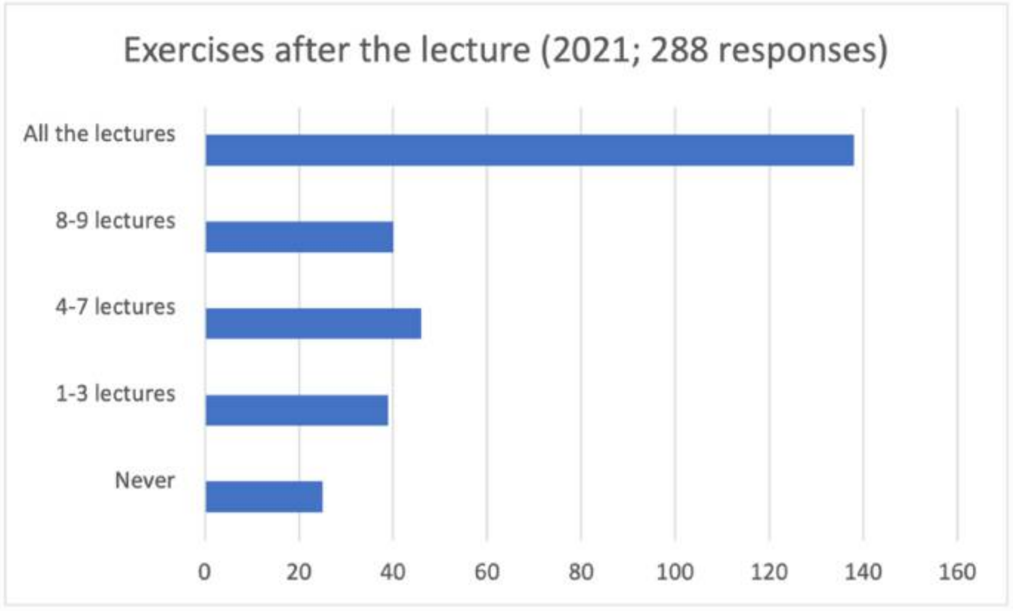
Students attended the lectures (well, as much as normal..) and asked better questions!



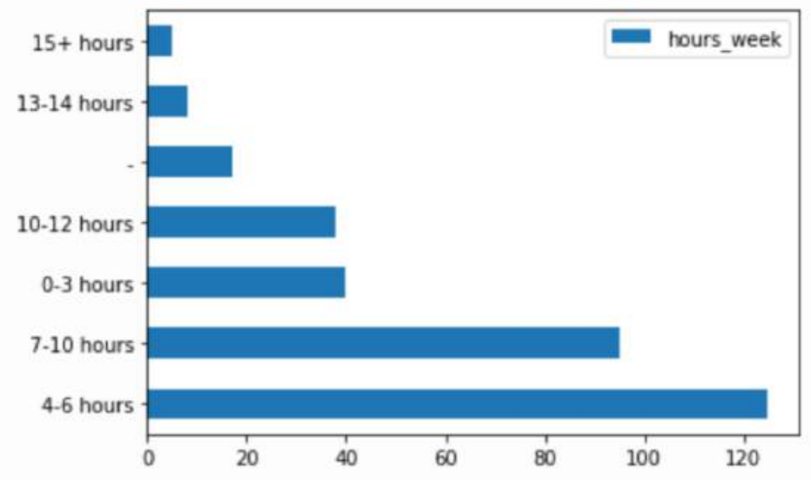


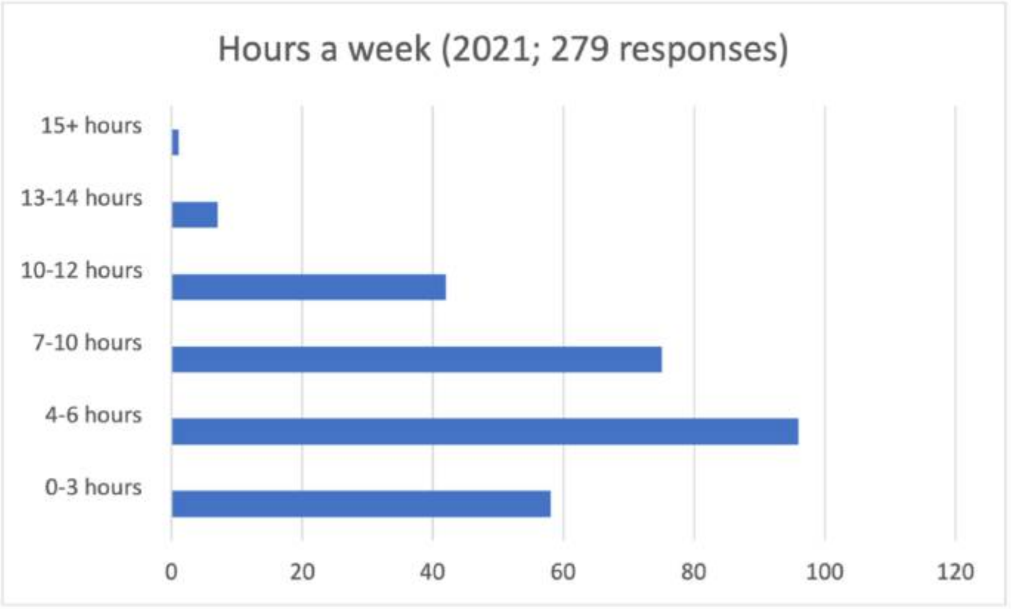
Focused exercises worked better than "large lab assignment"



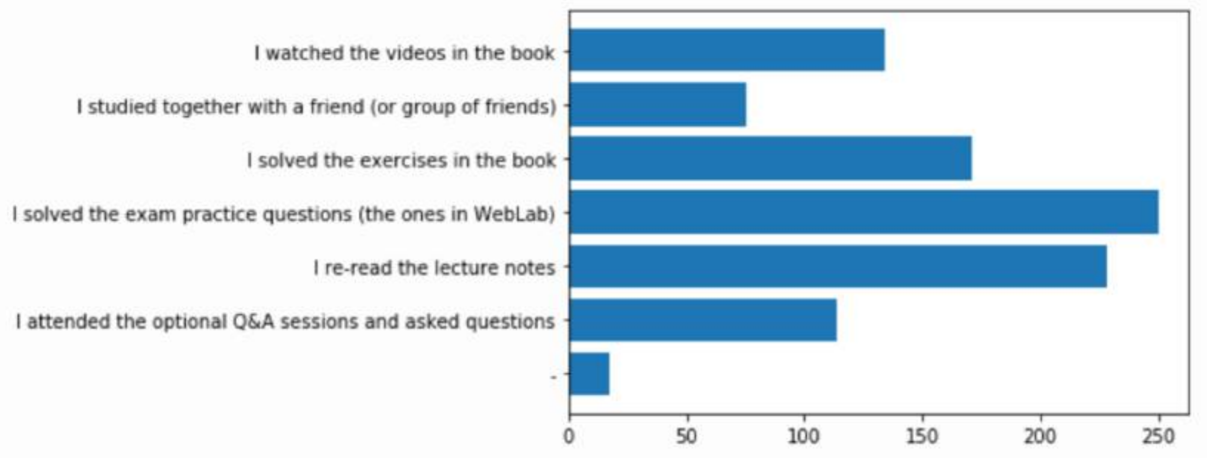


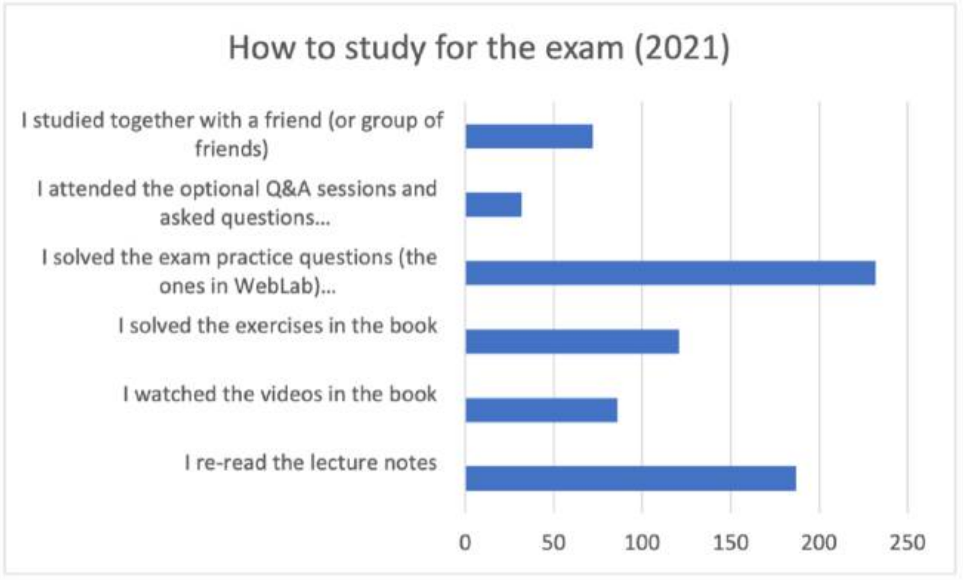
Students do not study the required number of hours





Do exam practice and read lectures are the popular study practices





Takeaways

- Creating lecture notes is a lot of effort, but it pays off.
- Trust that students will do their part. Do not have the perfect scenario in mind.
- Reserve time to answer questions, because there will be lots of them.

- https://www.mauricioaniche.com/blog/what-did-i-learn-from-flipping-my-classroom/

JUST
THANK
YOU!