Analyzing Student Work Patterns Using Programming Exercise Data

Jaime Spacco
Knox College

Paul Denny
University of Auckland

Brad Richards
University of Puget Sound

David Babcock
David Hovemeyer
James Moscola
York College of Pennsylvania

Robert Duvall
Duke University

SIGCSE 2015, March 4th-7th, Kansas City, Missouri, USA

Outline

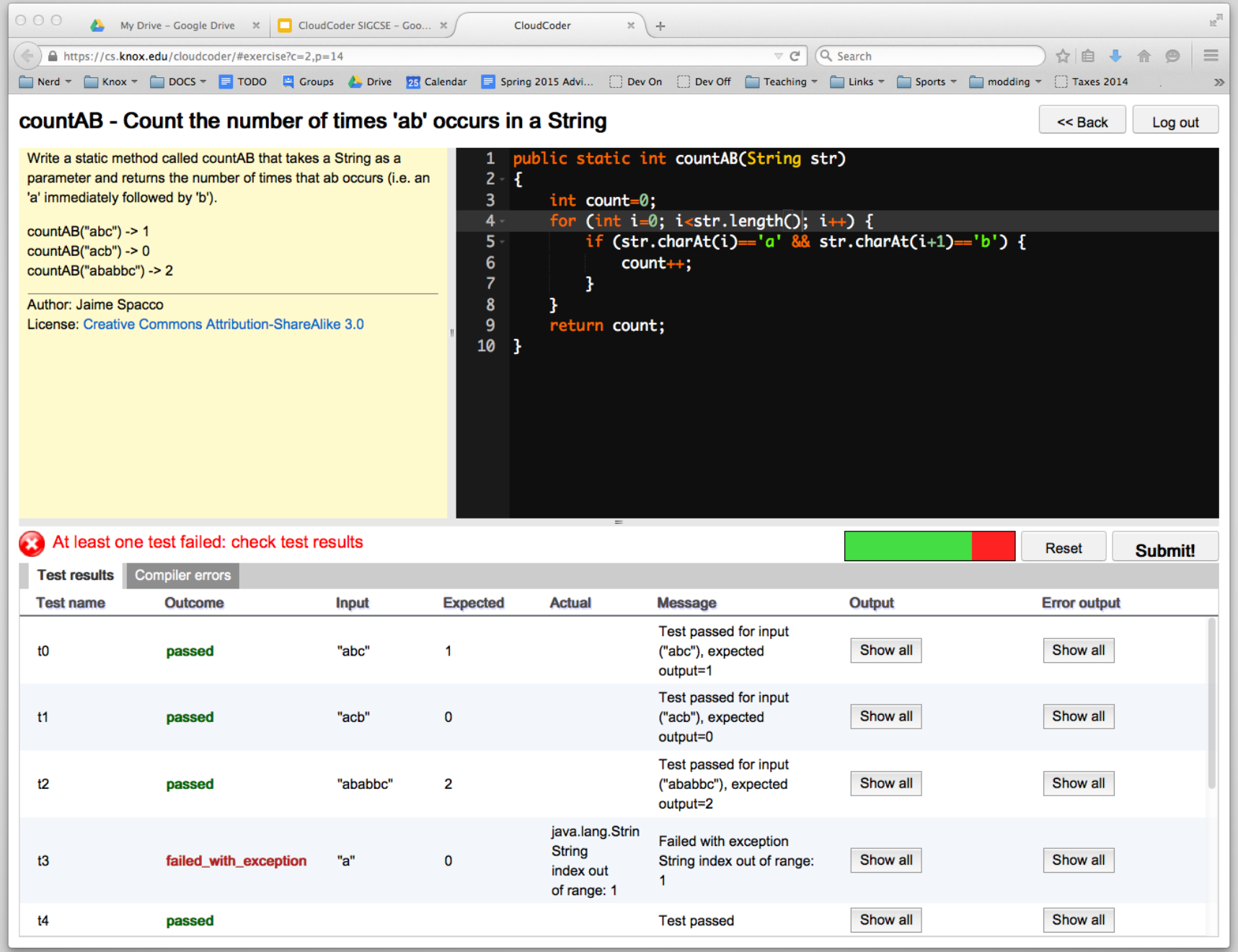
- CloudCoder
- Datasets
- Research questions
- Analysis of data, possible interpretations
- Conclusions

CloudCoder

- Open source web-based programming exercise system inspired by† CodingBat
- Exercises in Java, Python, C, C++, Ruby
- Students write short functions/programs
  - the opposite of Nifty
- Test cases used to judge correctness
- Automated feedback: useful for allowing students to practice outside of class
- Web: http://cloudcoder.org

†i.e., rip-off of

CloudCoder screenshot



CloudCoder long term goals

- Maximize opportunities for students to practice and develop skills
- Detect students who are struggling
- Early warning system for at-risk students
- *Help* students who are struggling
  - Hint generation!

TutorGen, Inc.

A Revolution in Learning

CloudCoder exercise repository

- Repository of permissively licensed (CC-BY-SA) exercises, contributions welcome
  - https://cloudcoder.org/repo
- Exercises are easy to "plug in" to an arbitrary course
  - They don't require much context
  - They don't have explicit dependencies on specific lectures/topics
- The exercise format is simple/open
  - Can be used with other systems

Fine-grained data collection

- Novel feature of CloudCoder: each edit event and submission recorded in database
  - With millisecond-resolution timestamps
  - Edit events are typically at keystroke level
  - Submission events record passed/failed tests
- Provides a very detailed (too detailed?) window into how students work

What do we do with this data?

This paper: analyze the data to see what interesting phenomena can be seen

Datasets

School	Term	Course	Language	# stud.	total # exercises	avg. # started	avg. # finished
York	Spring 2013	CS 1	C	133	62	37	33.9
York	Spring 2014	CS 1	C	86	53	30.2	27.4
Auckland	Fall 2013	Programming for engineers	C	740	12	11.7	11.6
Duke	Fall 2013	CS 1	Python	233	62	44.2	38.2
Duke	Spring 2014	CS 1	Python	194	55	45.2	42.2

- One assignment at Auckland worth 2% of final grade
  - Half of the course in C, half in Matlab
  - No CloudCoder exercises in Matlab
- Not graded at York
  - Used for both outside-class reading exercises, in-class “flipped class” exercises
- Required weekly exercises at Duke worth 10% of grade

Research questions

- Does work on exercises predict success?
- Is effort correlated with success?
- Can we find evidence of students struggling?
- Can we characterize relationship between exercise difficulty and required effort?

Do exercises predict exam success?

		Auckland	York 2013	York 2014	Duke 2013	Duke 2014
num. completed vs. final exam	p-value	< 0.001*	0.083	0.004*	< 0.001*	< 0.001*
	R^2	0.089	0.030	0.119	0.231	0.386
num. attempted vs. final exam	p-value	< 0.001*	0.370	0.006*	< 0.001*	< 0.001*
	R^2	0.073	0.008	0.106	0.194	0.382
pct. completed vs. final exam	p-value	< 0.001*	< 0.001*	0.041*	< 0.001*	< 0.001*
	R^2	0.082	0.138	0.06	0.149	0.295

Linear regressions predicting final exam scores with CloudCoder exercises attempted, completed, and percent completed.

- Statistically significant, but weak relationship at Auckland and York. Stronger relationship at Duke.
- Of course, we have no idea if this is causation or correlation.

What do these results mean?

- How exercises are integrated into course probably matters
  - Required exercises may be more predictive
  - Weekly exercises may be more predictive than one-off assignments
- There may be more to the story if we drill down further
  - Are some exercises more predictive?
  - Contact us with ideas
    - We can always use more co-authors

Effort vs. difficulty

		Auckland	York 2013	York 2014	Duke 2013	Duke 2014
avg. num. sessions vs. avg. best score	p-value	$< 0.001^*$	$< 0.001^*$	$< 0.001^*$	$< 0.001^*$	$< 0.001^*$
	R^2	0.702	0.770	0.593	0.218	0.654
pct. compilable subs. vs avg. best score	p-value	0.525	0.120	0.109	0.197	0.180
	R^2	0.042	0.040	0.050	0.028	0.034

Linear regressions predicting average best score on exercises based on average number of work sessions and percentages of submissions that compiled.

Effort vs. difficulty

		Auckland	York 2013	York 2014	Duke 2013	Duke 2014
avg. num. sessions vs. avg. best score	p-value	$< 0.001^*$	$< 0.001^*$	$< 0.001^*$	$< 0.001^*$	$< 0.001^*$
	R^2	0.702	0.770	0.593	0.218	0.654
pct. compilable subs. vs avg. best score	p-value	0.525	0.120	0.109	0.197	0.180
	R^2	0.042	0.040	0.050	0.028	0.034

Linear regressions predicting average best score on exercises based on average number of work sessions and percentages of submissions that compiled.

- Relatively strong *negative* correlation between number of sessions and average best score
  - Harder exercises (lower average best score) require more work

Effort vs. difficulty

		Auckland	York 2013	York 2014	Duke 2013	Duke 2014
avg. num. sessions vs. avg. best score	p-value	$< 0.001^*$	$< 0.001^*$	$< 0.001^*$	$< 0.001^*$	$< 0.001^*$
	R^2	0.702	0.770	0.593	0.218	0.654
pct. compilable subs. vs avg. best score	p-value	0.525	0.120	0.109	0.197	0.180
	R^2	0.042	0.040	0.050	0.028	0.034

Linear regressions predicting average best score on exercises based on average number of work sessions and percentages of submissions that compiled.

- No significant correlation between percentage of compilable submissions and average best score
  - Harder exercises don't seem to correlate with more syntax errors

What do these results mean?

- Some students struggle and need multiple work sessions
- Logic seems to be more difficult than syntax
  - This fits the intuitions of instructors
- What does “struggling” look like?

Hypotheses

Struggling students will:

- take more time
  - total time in minutes
- submit more often due to unproductive trial-and-error programming
  - number of submissions per minute

Students struggling

		Auckland	York 2013	York 2014	Duke 2013	Duke 2014
total time in mins. vs. pct. compilable subs.	p-value	$< 0.001^*$	$< 0.001^*$	$< 0.001^*$	$< 0.001^*$	$< 0.001^*$
	R^2	0.090	0.007	0.008	0.009	0.019
avg. # subs./min. vs. % compilable subs.	p-value	$< 0.001^*$	$< 0.001^*$	$< 0.001^*$	$< 0.001^*$	$< 0.001^*$
	R^2	0.195	0.026	0.020	0.024	0.020
total time in mins. vs. best score	p-value	0.002^*	$< 0.001^*$	$< 0.001^*$	0.034^*	0.417
	R^2	0.001	0.004	0.015	< 0.001	< 0.001
avg. # subs./min. vs. best score	p-value	0.507	0.072	0.405	0.017^*	0.435
	R^2	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001

Correlate effort/activity (total time spent, submissions/minute) with success (percentage of successful compilations, best score)

Students struggling

		Auckland	York 2013	York 2014	Duke 2013	Duke 2014
total time in mins. vs. pct. compilable subs.	p-value	$< 0.001^*$	$< 0.001^*$	$< 0.001^*$	$< 0.001^*$	$< 0.001^*$
	R^2	0.090	0.007	0.008	0.009	0.019
avg. # subs./min. vs. % compilable subs.	p-value	$< 0.001^*$	$< 0.001^*$	$< 0.001^*$	$< 0.001^*$	$< 0.001^*$
	R^2	0.195	0.026	0.020	0.024	0.020
total time in mins. vs. best score	p-value	0.002^*	$< 0.001^*$	$< 0.001^*$	0.034^*	0.417
	R^2	0.001	0.004	0.015	< 0.001	< 0.001
avg. # subs./min. vs. best score	p-value	0.507	0.072	0.405	0.017^*	0.435
	R^2	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001

Correlate effort/activity (total time spent, submissions/minute) with success (percentage of successful compilations, best score)

- Significant but extremely weak *negative* correlation between total time and subs/min vs. percent that compile
  - all relationships are in the right direction

Students struggling

		Auckland	York 2013	York 2014	Duke 2013	Duke 2014
total time in mins. vs. pct. compilable subs.	p-value	$< 0.001^*$	$< 0.001^*$	$< 0.001^*$	$< 0.001^*$	$< 0.001^*$
	R^2	0.090	0.007	0.008	0.009	0.019
avg. # subs./min. vs. % compilable subs.	p-value	$< 0.001^*$	$< 0.001^*$	$< 0.001^*$	$< 0.001^*$	$< 0.001^*$
	R^2	0.195	0.026	0.020	0.024	0.020
total time in mins. vs. best score	p-value	0.002^*	$< 0.001^*$	$< 0.001^*$	0.034^*	0.417
	R^2	0.001	0.004	0.015	< 0.001	< 0.001
avg. # subs./min. vs. best score	p-value	0.507	0.072	0.405	0.017^*	0.435
	R^2	< 0.001	< 0.001	< 0.001	< 0.001	< 0.001

Correlate effort/activity (total time spent, submissions/minute) with success (percentage of successful compilations, best score)

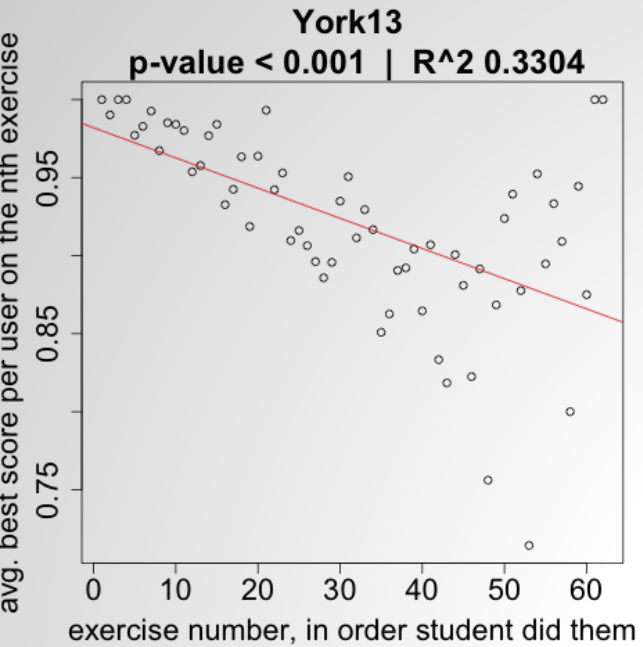
- Essentially no correlation between time and subs/min, and the best score

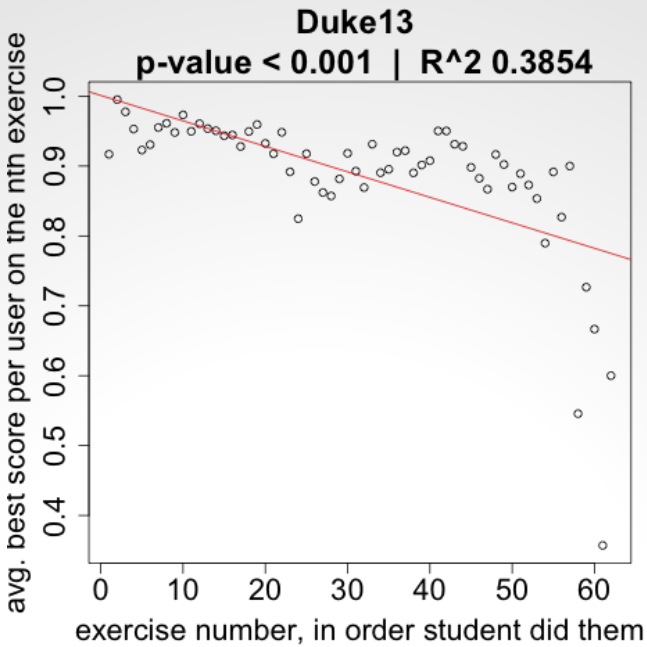
What do these results mean?

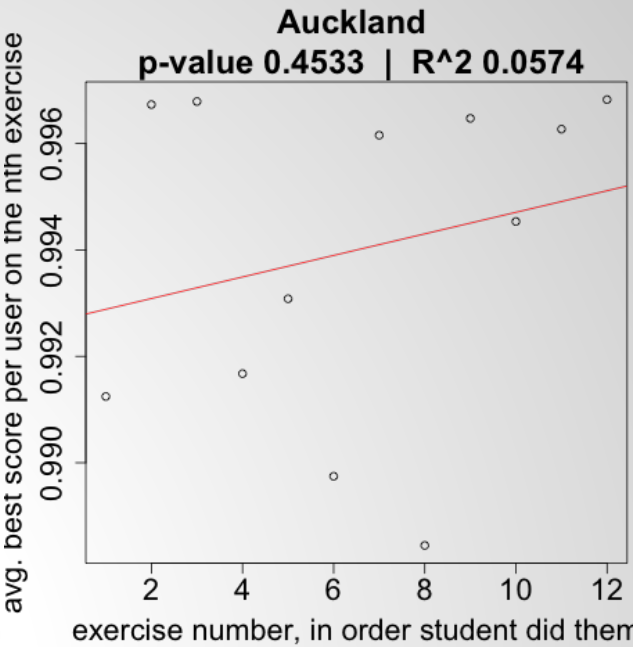
- The work patterns of a struggling student are (in general) more subtle than we expected
  - What else should we look for?

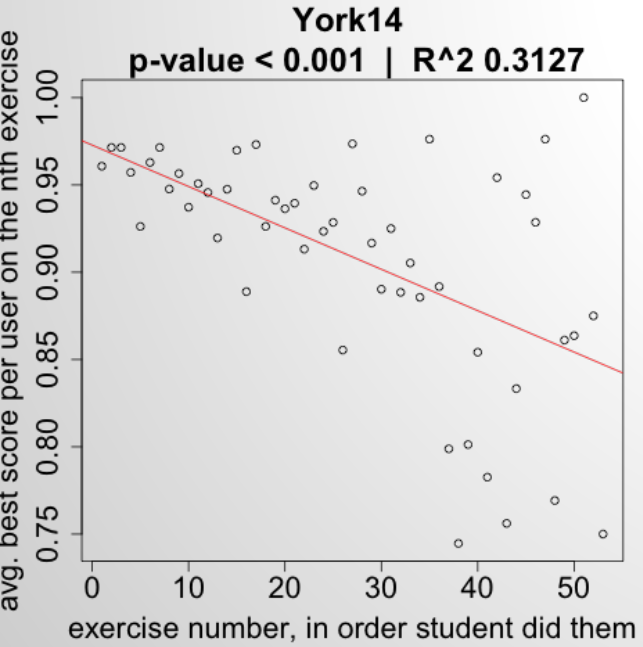
Do students improve?

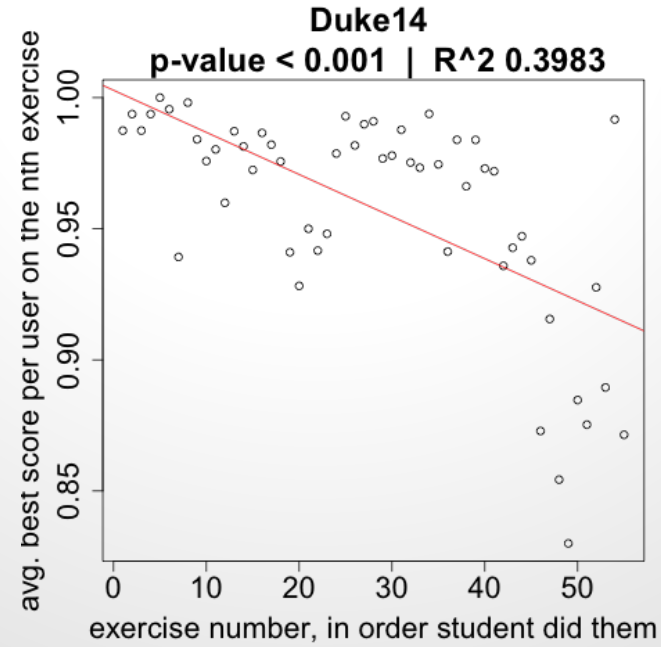
Look at average best score over time as exercises are assigned







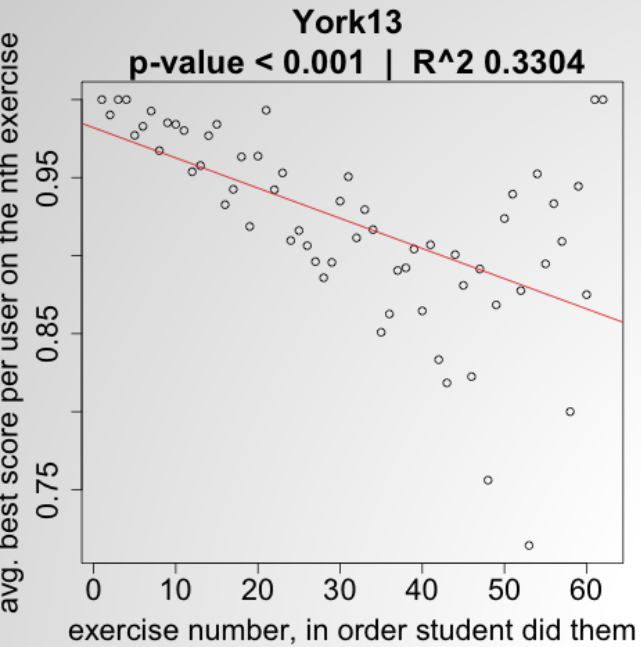


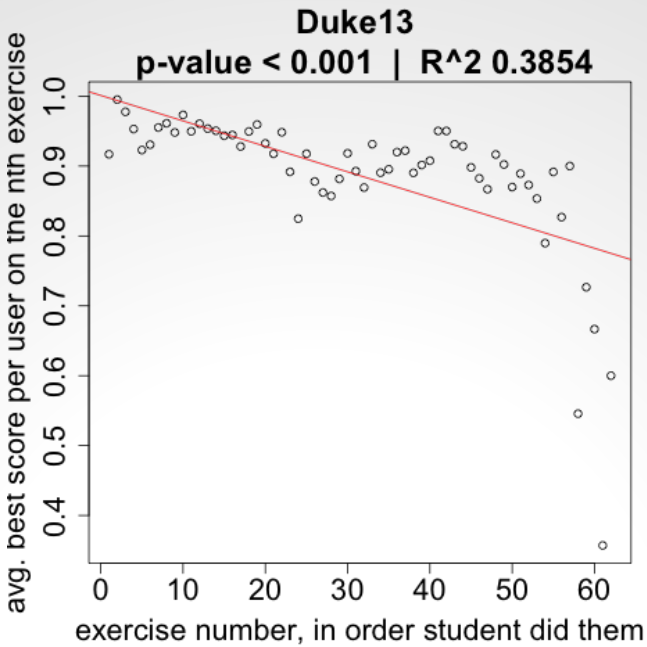


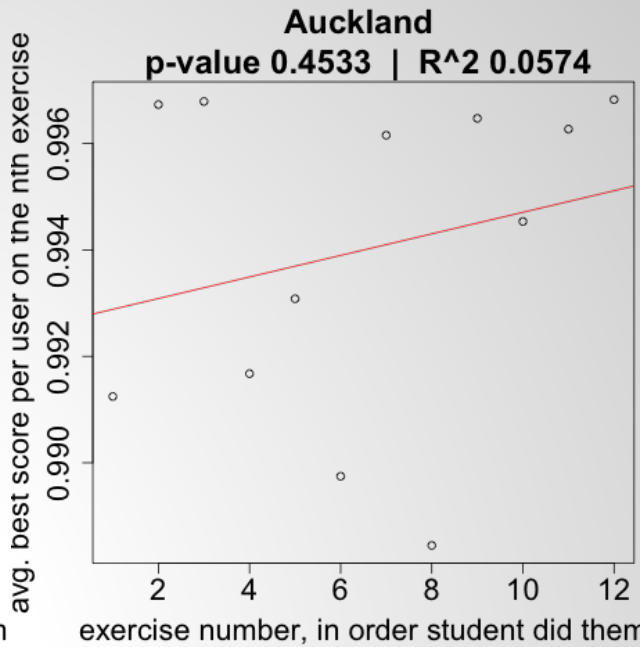
Do students get better as the term progresses?

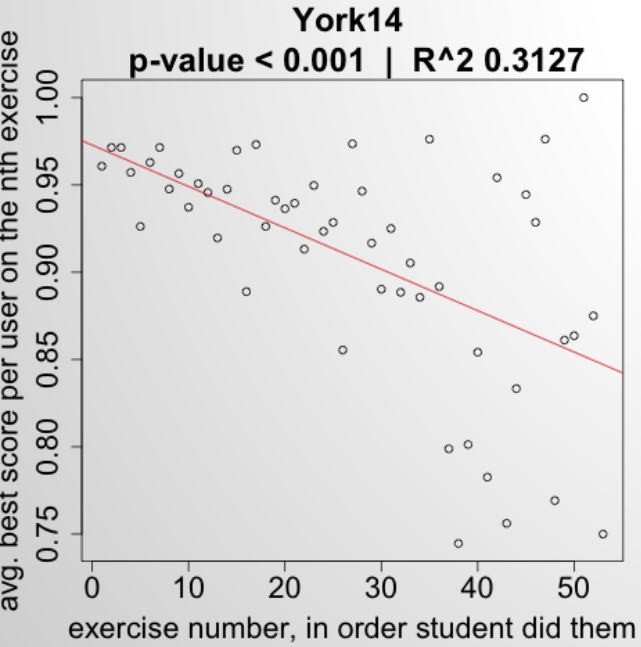
X axis: exercise #, in order student did them (students can do exercises on an assignment in any order)

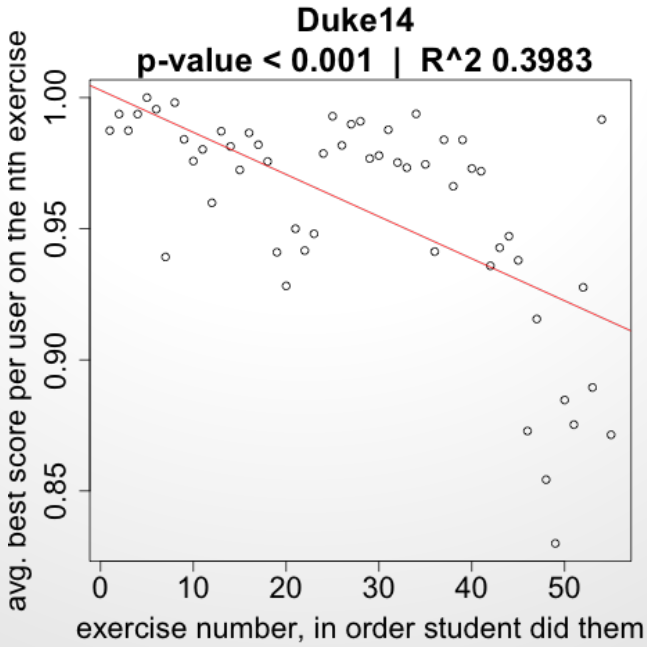
Y axis: average of the best score of each student attempting the exercise







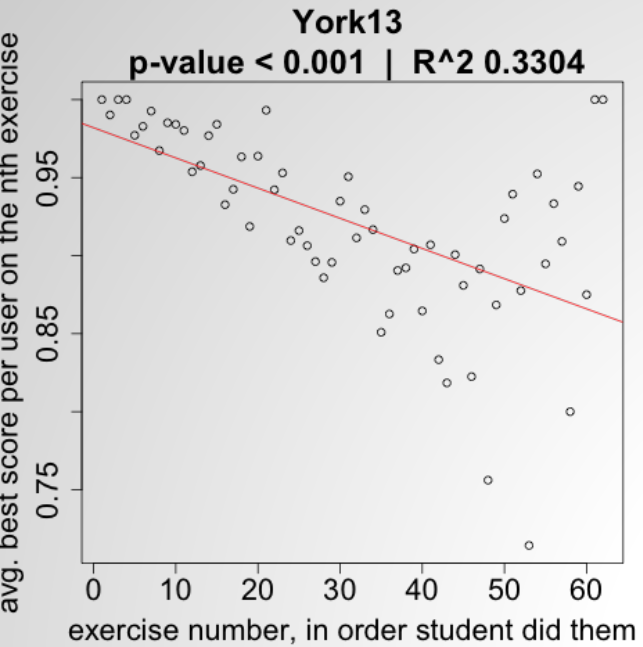


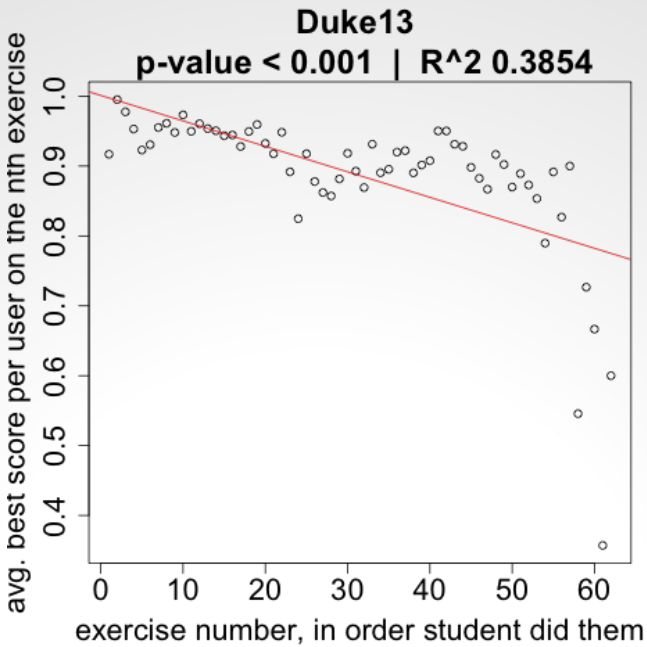


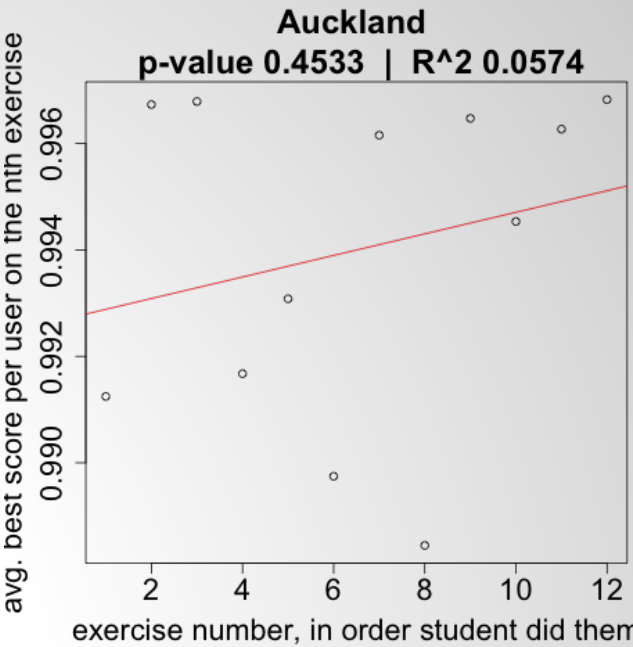
Do students get better as the term progresses?

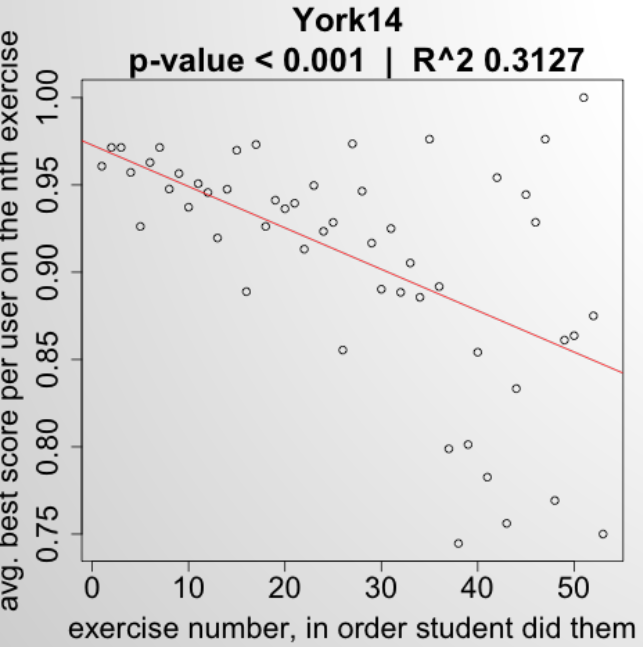
One possible answer:

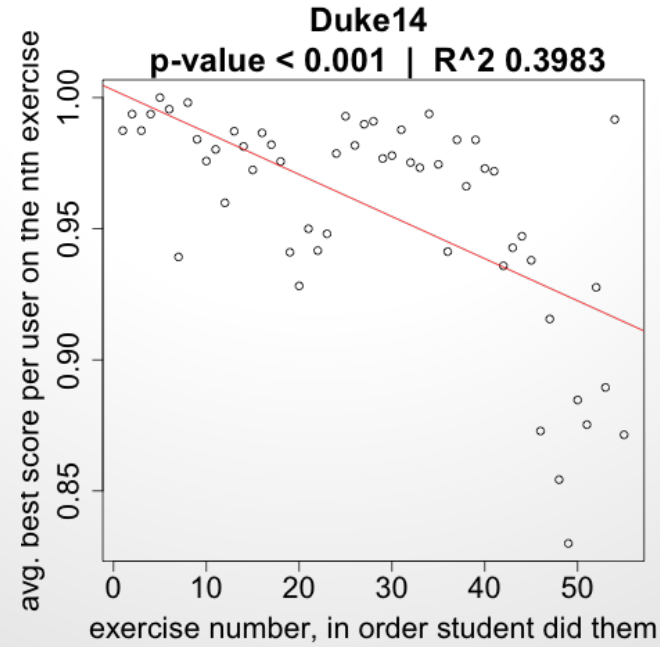
No. In fact, it looks like we make them worse!











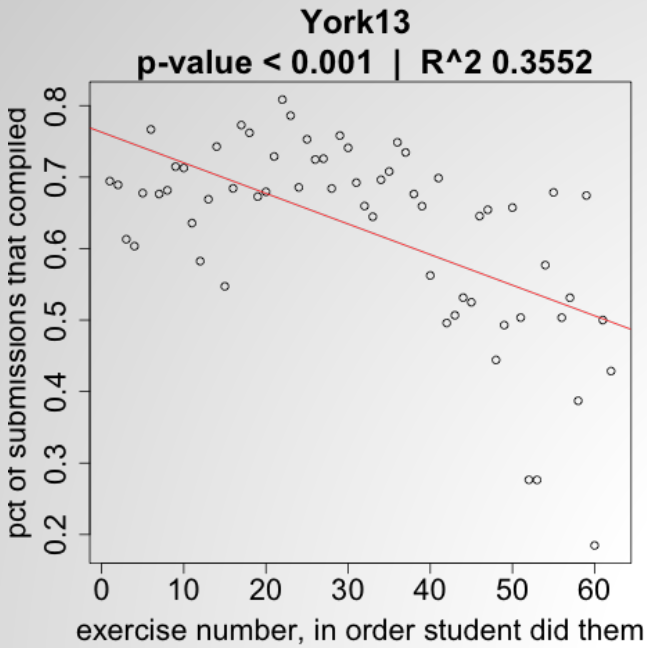
Do students get better as the term progresses?

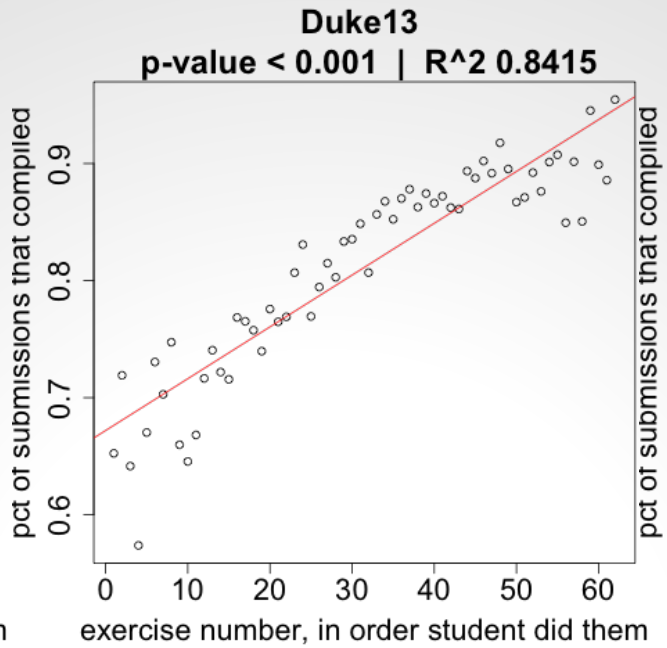
Another possible explanation:

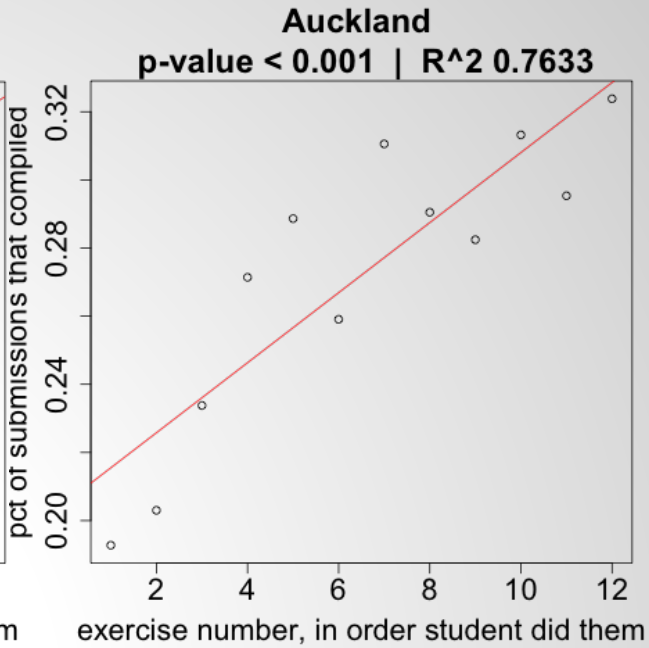
The exercises get more difficult as the term progresses.

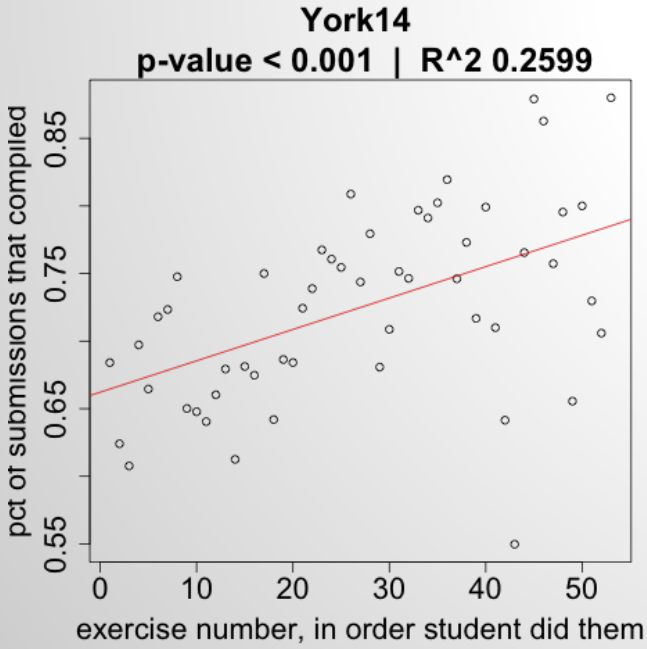
Does mastery of syntax improve?

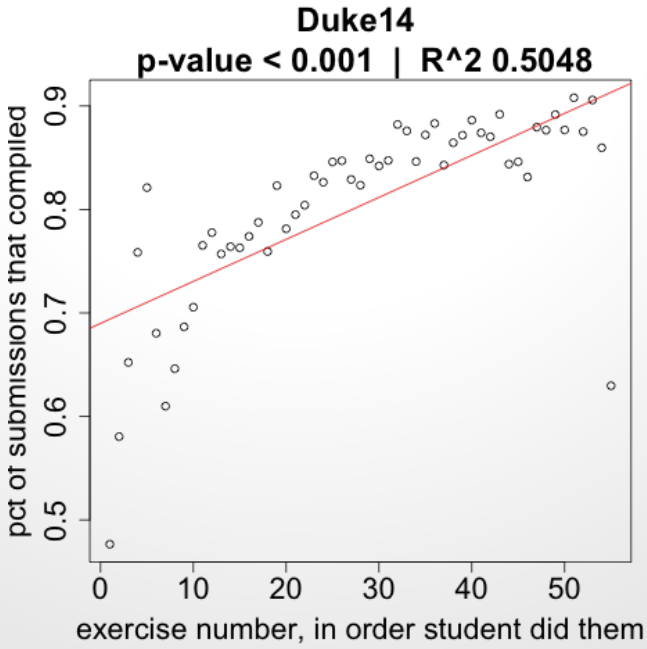
Do we see a greater percentage of compiling submissions as course progresses?







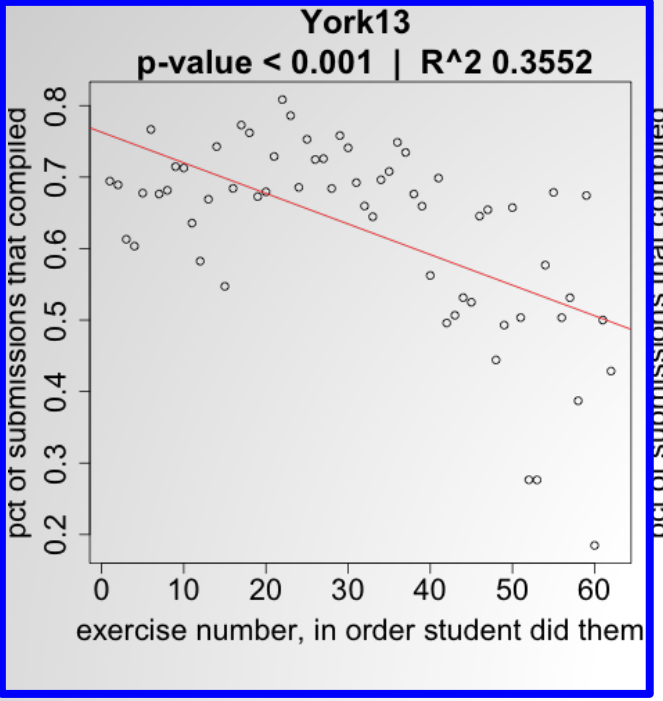


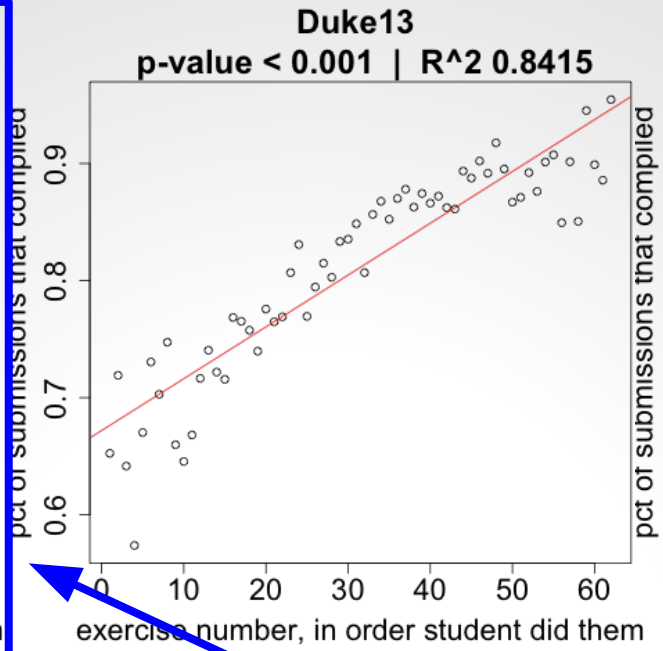


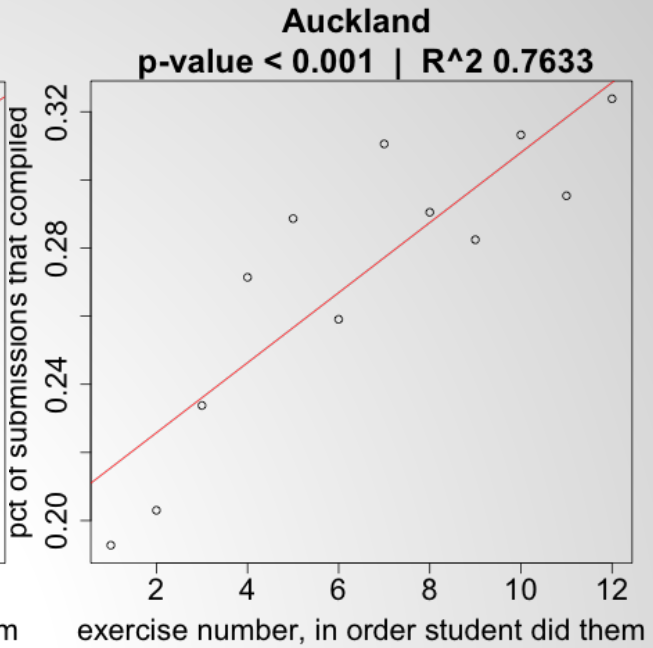
Does mastery of syntax improve?

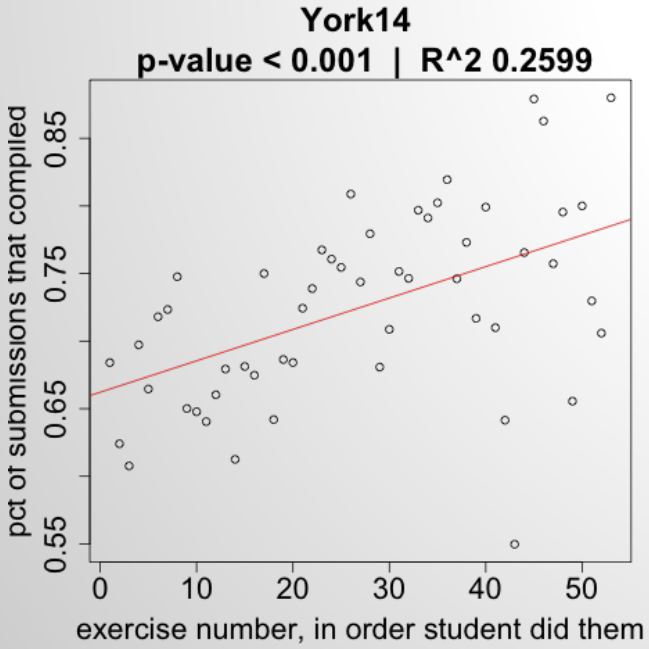
X-axis: Exercise #, in order students did them

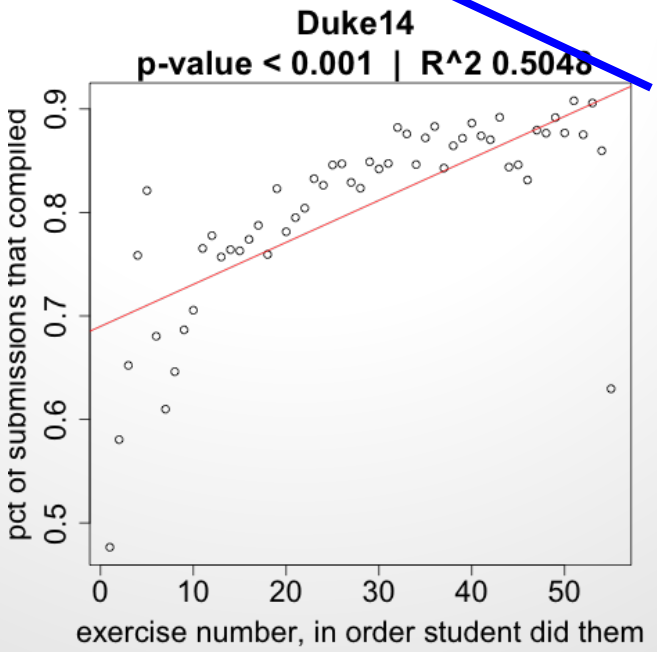
Y-axis: percent of submissions that compile







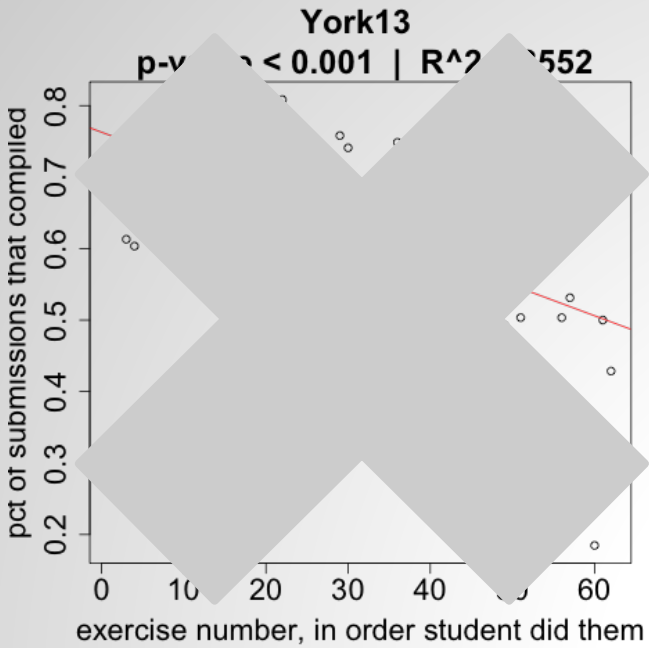


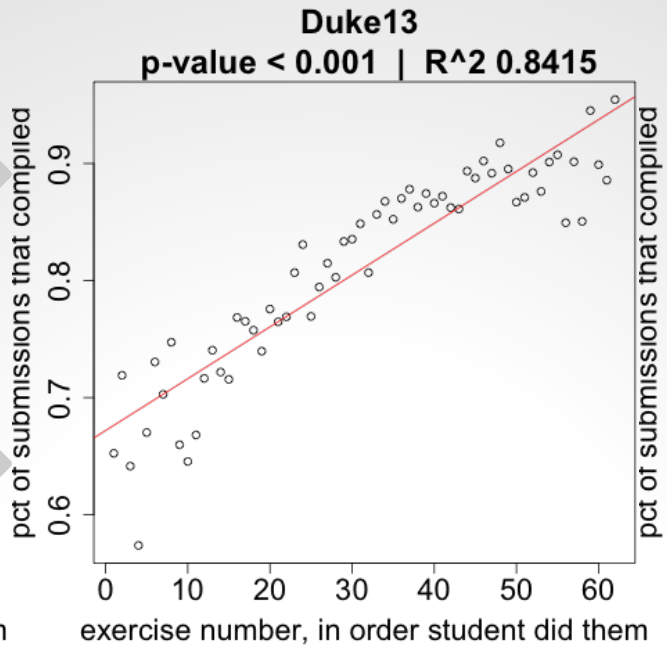


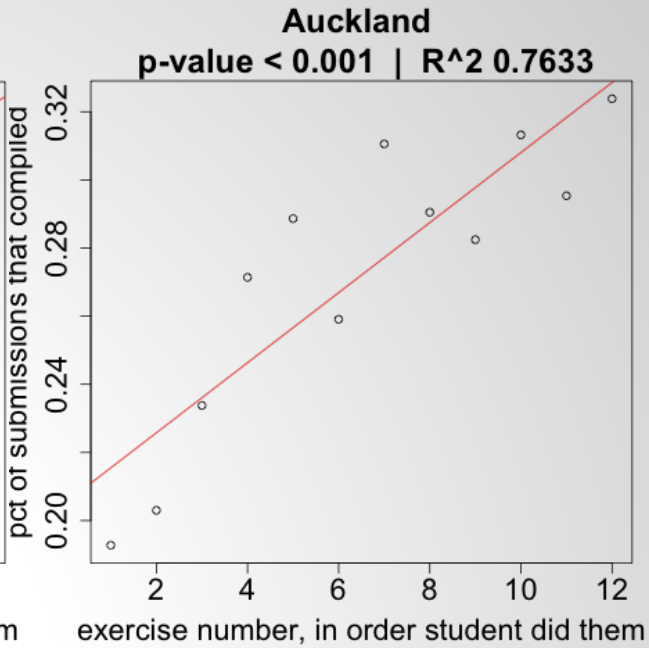
Some caveats:

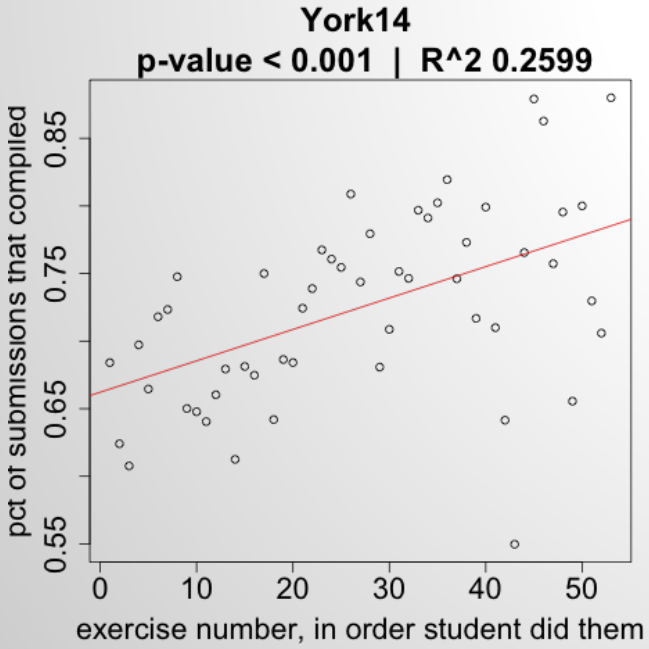
What the heck is happening here?

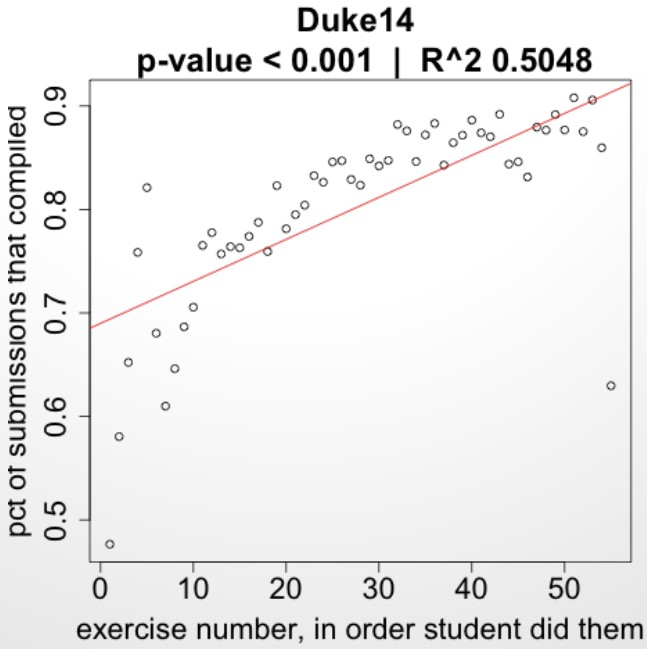
One possible explanation is that stronger students stopped doing the exercises over time (since they were optional).











Let's do what any good scientist would do!

This one is an outlier!

Beautiful trend for the rest of the data!

Conclusions

- Harder exercises require more effort
  - Duh!
- Struggling is not as easy to identify as we expected
  - Why? We have some ideas, no firm conclusions yet
- Syntax does not seem to be the primary difficulty
  - at least later in the course

Future work

- Does early performance on exercises predict success in course? [See Porter, Zingaro, and Lister, *Predicting student success using fine grain clicker data*, ICER 2014]
- Can we identify exercises that are particularly effective at reinforcing specific concepts and techniques?

Thank you!

Questions?