Effectiveness of the Automatic Reassessment and Relearning System for Short- and Long-term Mathematical Skill Retention

By

Manuel Santana Gonsalves

Advisor: Neil Heffernan

A Thesis submitted to the Faculty

Of the

WORCESTER POLYTECHNIC INSTITUTE

in partial fulfillment of the requirements for the

Degree of Master of Science

in

Data Science

By

May 2021

APPROVED:

Professor Mohamed Y. Eltabakh

Abstract

This thesis examines how the Automatic Reassessment and Relearning System (ARRS) affected long-term mathematical skill retention for individual students over multiple skills. 50 geometry students participated in a study lasting ten weeks. Students participating in the study completed a pre-test with 8 different skills, practiced until mastery for every skill, completed additional reassessment and relearning assignments for half of the skills in following weeks, and completed a 3-week and 9-week post-test for every skill. Using linear regression to estimate the significance ARRS assignments had in impacting students post-test scores we were not able to demonstrate that these additional assignments significantly improved student's skill retention in the short or long run.

1. Introduction

The main goal of education is for students to learn new concepts, but also retain them through the rest of their lives. To achieve this goal an intelligent tutoring system (ITS) called ASSISTments, utilizes 2 key principles: the testing effect and the spacing effect. The testing effect is a phenomenon where students retain more knowledge by being tested on material multiple times compared to just restudying the material. One study that examined students learning words in a new language demonstrated significantly better long-term retention of words when they were repeatedly tested on every word every time compared to just testing words they answered incorrectly in the previous test. The study also showed that there was no difference in long-term retention when changing the studying approach before each test (Karpicke & Roediger, 2008). Restudying the material is not as effective as testing because testing forces students to recall information and apply the skill in different contexts. ASSISTments constantly utilizes the testing effect by testing students using formative assessments. Formative assessments accomplish the goal of testing the students to gain the benefit of the testing effect, while providing important timely feedback to students and teachers. ASSISTments' has shown that constant formative assessments can significantly improve student learning (Roschelle, et al, 2016).

The spacing effect is where students are repeatedly exposed to material at different points in time, compared to being exposed once in a larger study period. One study found that by moving the homework for a certain topic to 5 days after instruction compared to the same day of instruction leads to much better long-term retention of the topic. They believe the spaced practice benefited the students because it was more difficult to recall information over a period of time causing "stronger conceptual connections and deeper comprehension" (Cadaret & Yates, 2018).

ASSISTments designed ARRS to encompass both the testing and spacing effect to improve student retention on previously mastered material. ARRS is built off a specific assignment type inside of ASSISTments, they call a skill builder. A skill builder is an assignment where the student will be randomly assigned problems from a large source of similar problems. If the student answers problems incorrectly they are given feedback in the form of hints or the correct answer. The student will continue to receive problems until they are able to get 3 in a row correct without any help. The completion of the assignment demonstrates a mastery of the skill.

After students have completed the skill builder, they will start the first reassessment level. 7 days after finishing the skill builder they will be given a reassessment assignment, which is an assignment containing one question pulled from the original source of problems. If the student answers incorrectly or must ask for help they will need to complete a relearning assignment, which is the original skill builder. After completing the relearning assignment, they will receive another reassessment assignment after 7 days. If the student correctly answers the question in the reassessment, they will move to the next reassessment level. At level two the student will receive the reassessment after 14 days of the previous level completion or a relearning assignment completion. There are 4 levels in total and the spacing between reassessment assignments is 7, 14, 28, and 56 days at each level. Figure 1 is a visual representation of the process.

For example: if a student completed their 1st reassessment successfully after 7 days, they would receive the next reassessment in 14 days. If the student could not successfully complete the 2nd reassessment, they would complete the relearning assignment and then attempt the reassessment assignment again in 14 days. After successful completion of the 2nd level reassessment assignment they would receive the 3rd level reassessment assignment in 28 days.

2. Related Work

Previous studies have been conducted to test the effectiveness of ARRS. The first study addresses the difference in predictive ability of short-term skill retention and long-term skill retention and the benefit of extra information from student response data from reassessment and relearning assignments in predicting long-term skill retention. The study included 128 students and 33 skills. The skills were separated into 2 groups A and B. Students were broken up into two groups where both groups completed all the skill builders, then group 1 completed ARRS for the skills in group A, and group 2 completed ARRS the skills in group B. All students took a post-test with all 33 skills 6 months later.

To predict the short-term skill retention the first reassessment assignment was used as a post-test score and the long-term skill retention used the 6-month post-test. Using logistic regression, the study demonstrated that short-term retention was easier to predict than long-term retention when using the same input. The study also found that by adding information from the students' reassessments and relearning assignments the predictive power of the logistic regression could be improved again (Wang & Heffernan, 2011). However, the study did not look at how the positive or negative effect that the ARRS features had on student retention.

The second study a few years later analyzed the effect of ARRS on long-term retention for individual students across specific mathematical skills. This study included 97 8th grade students and 32 skills. The study was set up in a similar way to the first. The skills were separated into Set A and Set B with the average difficulty in each set being of similar difficulty. All students completed every skill builder and were assigned to do ARRS for skills in either Set A or Set B. The pre-test for each skill was the first question in the skill builder assignment. An end of year post-test was given to all the students that contained one question for every skill.

To determine the benefit that ARRS produced compared to mastery-only of the skill 3 tests were conducted. The first was a simple difference of means test that found no significant difference in pre-test scores between the ARRS and mastery-only, but a significant effect on post-test scores between the 2 groups. The second test performed regression analysis controlling for pre-test scores and found that the ARRS condition was significantly higher than the mastery-only condition. The last test was another regression analysis to determine how ARRS affected scores across skills of varying difficulty. Again, the study found a significant increase in the ARRS condition compared to the mastery-only condition (Soffer, et al., 2014).

3. Previous Work

This thesis will be using the data collected in a study in 2019. The study was conducted very similarly to the second study where skills were separated into two groups, the odds skills in group 1 and the even skills in group 2. The study included 32 geometry students who were paired by their first semester grades and then randomly assigned to one of the two groups. Groups 1 and 2 both completed an 8-question pre-test, where each question represented one skill they would be practicing. For the first 4 weeks, students completed 2 skills builders of relatively equal difficulty each week and completed the ARRS assignments for the group they were assigned to. Students in both groups completed a post-test 3-weeks after completing each skill builder and a final post-test containing 8 questions, one for each skill, 9 weeks after the initial pre-test. The study was conducted with a smaller class, but has the benefit of having 2 different post-tests, so we can measure the benefit of ARRS in short-term skill retention and long-term skill retention. Figure 2 displays a diagram of the study and Table 1 displays the skills being learned.

4. Methodology

We collected the data from the ASSISTments platform where the students completed all the assignments. The study began with 8 skills and 32 students. The wrong group completed ARRS assignments for skill 6 leaving some students to complete ARRS assignments for 5 skills and the other students completed ARRS assignments for 3 skills. We removed both skill 5 and 6 from the analysis because each pair of skills had a different difficulty to the other skill pairs and could lead to skewed results. 6 students were also removed as they had either not participated in ARRS assignments for 3 skills or failed to complete a skill builder assignment for one of the skills. We also created a second data set from the clean one that only looked at students who completed at least 3 reassessment assignments.

For the analysis of the study a total of 6 were used. The skills were paired together based on their difficulty and then split between two groups A and B. Each of the 26 students completed skill builders for every skill and ARRS assignments for either skills in group A or group B. This allowed students to engage in both conditions.

We used a regression analysis to analyze the results. We used the skill_id, student_id, practice condition (ARRS, mastery-only) and the pre-test as independent variables and the post-tests (3-week post-test, 9-week post-test) as dependent variables. We used the 3-week post-test as a measure of short-term skill retention and the 9-week test as the measure for long-term skill retention.

5. Data Collection

The first step in being able to conduct the analysis for the study was putting together the data set that would be analyzed. When the data was originally collected the only data that was collected was how each student performed on the pre- and post-tests and whether they were in the ARRS condition or not. For our analysis we needed more information about how the students completed their assignments, so we quired the ASSISTments database to gather the information. Figure 3 represents a model of the entire ASSISTments database that needs to be queried.

The first part was determining the information that we needed for the study and the best format in which to analyze it. The dataset used in the analysis has one row for each student skill pair. This means each student is represented 6 times in the data set, one for each skill. Every row in the dataset includes:

Name of variable	Description
Name	Name of the group the student was in
Student_id	The students id
User_id	The assistments account id
Skill_id	The id of the skill being tested
Skill_name	The name of the skill being tested
Pre_problem_id	The problem id of the pre-test
Pre_test_score	Score on the pre-test
Delayed_problem_id	The problem id for the 3-week post-test
Delayed_test_score	Score on the 3-week post-test
Post_problem_id	The problem id for the 9-week post-test
Post_test_score	Score on the 9-week post-test
Problems_completed in_skill_builder	Number of problems completed in the original skill builder
Mastery_status	the status of the original skill builder assignment
Avg_reassessment_score	Average score on reassessment assignments
Reassessments_completed	The number of reassessment assignments completed
Completed_reassessment_level	The highest completed reassessment level
Relearnings_attempted	Number of relearning assignments attempted
Relearnings_completed	Number of relearning assignments completed
Arrs	Whether the student completed arrs assignments for the skill
Student_reassessment_record_id	The id of the reassessment record

As seen in Figure 3 the ASSISTments database is very large and contains a lot of information that does not pertain to this study. Figure 4 is a reduced model of the ASSISTments database containing only the tables pertinent to the study. To be able to connect the skills, problems, and skill builder sequences we added one more table, which can be found as Query 1 in the appendix. This table was built manually

by looking at all the assignments in the class and determining which problems and sequences corresponded with each skill. Most of the assignment, problem, and sequence information was found by looking at the assignments in ASSISTments online and inspecting the page. For some strange reason any of the skill builders that students completed ARRS for were not connected to the classroom. This means they were not displayed online and did not have a class id associated with them in the database. These assignments had to be found by combing through the database.

Query 2 is the final query that was used to compile the dataset. Throughout Query 2 you will notice that there are specific ids that are used for assignments, problems, and sequences. We used specific assignment and problem ids so that we could find and isolate problems in the pre- and post-test assignments to one row. We used sequence ids to gather information about the skill builder and ARRS assignments. The lists of ids in Query 2 correspond with ids in Query 1.

Query 2 is broken down into 4 general parts which are separated by a line of dashes. The first part is getting the students from correct class and assigning them to the correct groups. The second part of the query is gathering and grouping the results from the pre- and post-tests. The third part of the query found information about how students completed their ARRS assignments. The final part of the query gathered information on how students completed each skill builder. The final part of the query uses a union and similar code due to the issue mentioned earlier about the skill builders with ARRS attached were not connected to the class.

6. Data cleaning

While the data was collected directly from the ASSISTments database some cleaning was still needed to ensure the analysis was completed correctly. To complete the cleaning and the analysis we used R-studio and R code. The first step was loading the data into R-studio from the csv downloaded from the result of Query 2. All the values in the dataset were strings when loaded in. We used the transform function to convert variables from strings to their appropriate data type. We then filled in the null values with the correct replacement value. Once this step was complete, we had a data set containing information about every student in the study.

As we are conducting analysis for a study, we also needed to ensure that the students in the data set completed all 6 assignments. To do this we first filtered out all the students who did not have 3 ARRS records and then filtered out the students who didn't have ARRS for half of their records. As we were taking a precursory look at the data, we noticed that there were some students who were not completing their ARRS assignments. We decided to create another data with only students who completed at least 3 reassessment assignments. Both data sets created were analyzed in the exact same way.

After completing the cleaning, we were left with 2 data sets. The first set contained data on 26 students, 13 in each group, with a total of 156 observations. The second set contained data on 19 students, 11 in group 1 and 8 in group 2, with a total of 114 observations.

7. Model Selection

To conduct our analysis, we finally decided on ordinary least squares regression with Robust Standard Errors for 3 reasons. The first is that this is a study. We needed to determine if the completion of ARRS assignments changed student learning significantly. This removes many complicated models, such as neural networks, that lose track of how certain features affect the model.

Secondly, we noticed our cleaned data sets were not rich in observations. We wanted to avoid models that would have more parameters than there were instances of data. This could lead to drastic over fitting in the model. This removed more complicated models such as random forest which can calculate feature importance, but also has a lot of parameters.

Lastly, ordinary least squares regression would likely have heteroscedastic errors because the scores for the post-tests were binary variables. We choose a model that would correct the heteroscedastic errors.

8. Results

This study examined how beneficial the ARRS condition was on student skill retention compared to the mastery-only condition. We conducted ordinary least squares regression with Robust Standard Errors for both the 3-week post-test and 9-week post-test. We used the skill id, the student id, practice condition (ARRS, mastery-only), and the pre-test scores as the predictors. We analyzed both data sets the same way.

For the first data set, students who completed 6 assignments, the short-term model using skill id, student id, condition, and pre-test scores as predictors for the 3-week post-test score we found the variables explained 20.1 % of the variance ($F(32,123)=6.02$, $p<0.001$). The model also showed that the practice condition was not significant, $t(123) = 0.572$, $p=0.569$, in predicting short term skill retention. We also found no interaction between pre-test scores and 3-week post-test scores, $t(123)=1.015$, $p=0.312$.

For the second data set, students who completed at least 3 ARRS assignments, the short-term model using skill id, student id, condition, and pre-test scores as predictors for the 3-week post-test score we found the variables explained 16.6 % of the variance ($F(25,88)=3.796$, $p<0.001$). The model also showed that the practice condition was not significant, $t(88) = 0.901$, $p=0.370$, in predicting short term skill retention. We also found no interaction between pre-test scores and 3-week post-test scores, $t(88)= 0.108$, $p=0.914$.

For the first data set, the long-term model using skill id, student id, condition, and pre-test scores as predictors for the 9-week post-test we found the variables explained 32.3 % of the variance ($F(32,123)=8.22$, $p<0.001$). Again, the model showed no interaction between the practice condition and long-term skill retention, $t(123)=-0.162$, $p=0.872$. There was also no interaction between pre-test scores and 9-week post-test scores $t(123)=.301$, $p=0.764$.

For the second data set, the short-term model using skill id, student id, condition, and pre-test scores as predictors for the 9-week post-test score we found the variables explained 25.1 % of the variance ($F(25,88)=7.524$, $p<0.001$). The model also showed that the practice condition was not

significant, t(88) = 0.025, p=0.980, in predicting short term skill retention. We also found no interaction between pre-test scores and 9-week post-test scores, t(88)= -0.172, p=0.864.

9. Conclusions

From the results of our analysis we were not able to find any significant indication that ARRS was beneficial in students long or short-term retention of mathematical skills. This study was much smaller in the number of students, skills being tested, and the amount of time to complete compared to previous studies involving ARRS. While other studies of ARRS have shown significant improvements in skill retention, we hypothesize that the size of our study may have played a role in our conflicting findings. We suggest a similar study should be conducted involving more skills and more students to help clarify these discrepancies.

10. Current and Future Work

To be able to conduct this study again in the future I have been working with ASSISTments to develop the ARRS system on their new platform. The new platform is up and running with tens of thousands of more users than the original version. The new environment is a perfect place to recruit more teachers to get a larger sample of students.

The most important part in designing the new system was cleaning up how the information would be stored in the database. After having to search through the ASSISTments database and filter out redundant information to collect the data for the study, we knew there was some definite room for improvement. The new model is similar to the old one, as in there are still assignments and logs, but there is an addition of object property tables. With this table we can save teacher specific settings for ARRS and more importantly it allows us to attach information to specific assignments.

By attaching 3 fields of information to an assignment we can store all the information we need to connect ARRS assignments to the original skill builder and each other. The first property is the “assignment type”. This allows us to differentiate assignments like skill builders with ARRS from plain skill builders and reassessment assignments from relearning assignments. The second field is the parent assignment id. In the case of reassessment assignments, the parent assignment would always be the original skill builder, and for relearning assignments it would be the failed reassessment assignment. Lastly, we added assignment level so we can keep track of what reassessment level the student is at. If students failed a reassessment and needed to complete another one at the same level, we can determine which assignment came first by the date the student completed it. The rest of the information about the assignment is already saved in the assignment and assignment logs table.

By adding these three properties we were able to completely remove all the tables that stored ARRS data in the original version. This greatly reduces the amount of space that is consumed and the

number of redundancies in the database. This design also works perfectly for incorporating the new adaptive homework feature that is will eventually make its way to the ASSISTments platform.

11. References

Cadaret, C. N., & Yates, D. T. (2018). Retrieval practice in the form of online Homework improved information retention more when SPACED 5 days rather than 1 day after class in TWO Physiology courses. *Advances in Physiology Education, 42*(2), 305-310. doi:10.1152/advan.00104.2017

Karpicke, J. D., & Roediger, H. L. (2008). The critical importance of retrieval for learning.*Science, 319*(5865), 966-968. doi:10.1126/science.1152408

Roschelle, J., Feng, M., Murphy, R. F., & Mason, C. A. (2016). Online Mathematics Homework Increases Student Achievement. *AERA Open*. https://doi.org/10.1177/2332858416673968

Soffer,-Goldstein, D., Das, V., Pellegrino, J., Goldman, S., Heffernan, N., Heffernan, C., & Dietz, K. (2014). Improving Long-term Retention of Mathematical Knowledge through Automatic Reassessment and Relearning. American Educational Research Association (AERA 2014) Conference. Division C - Learning and Instruction / Section 1c: Mathematics.

Wang, Y., & Heffernan, N. T. (2014). The effect of Automatic reassessment and Relearning on assessing student long-term knowledge in mathematics. *Intelligent Tutoring Systems,* 490-495. doi:10.1007/978-3-319-07221-0_61

12. Appendix

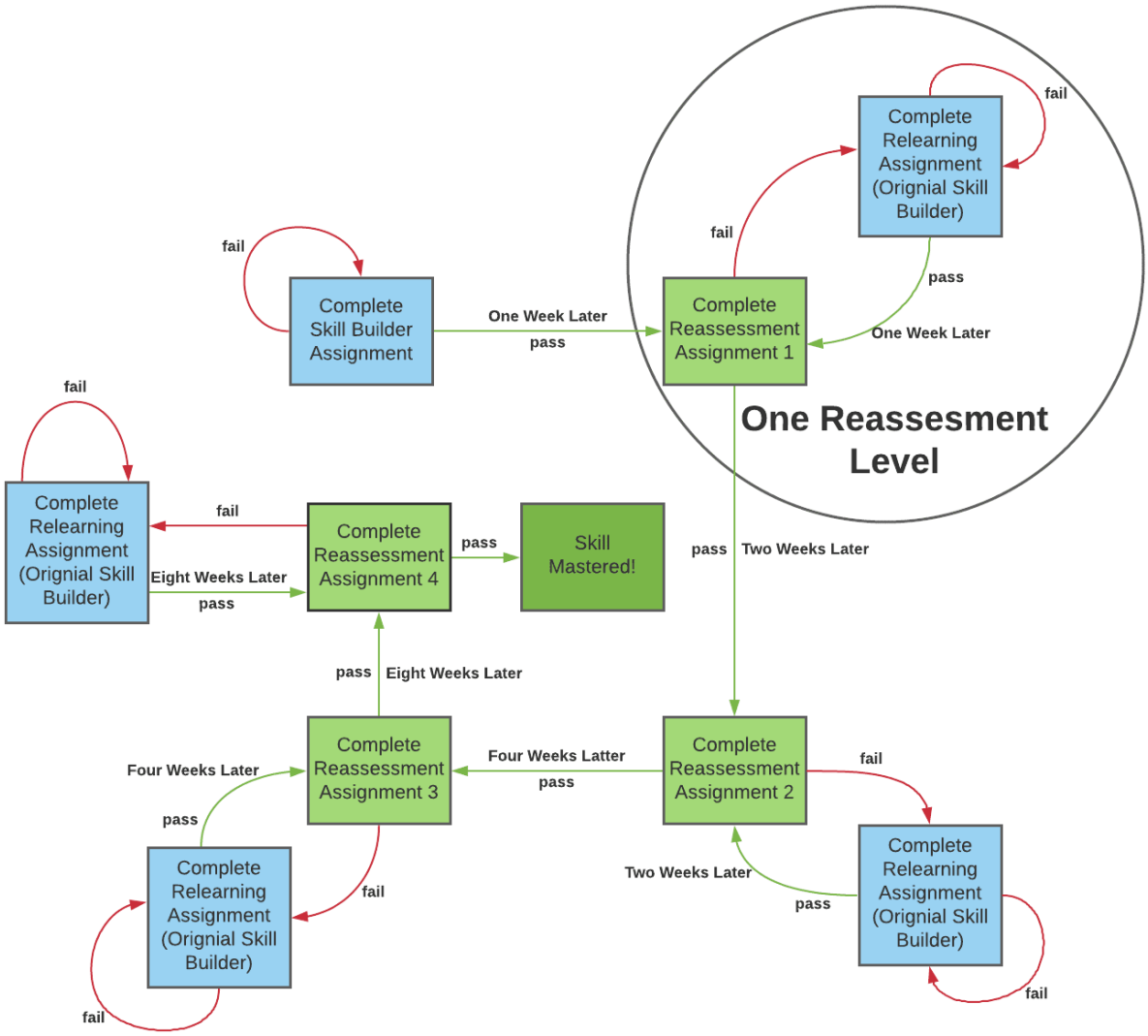

Figure 1: ARRS Diagram

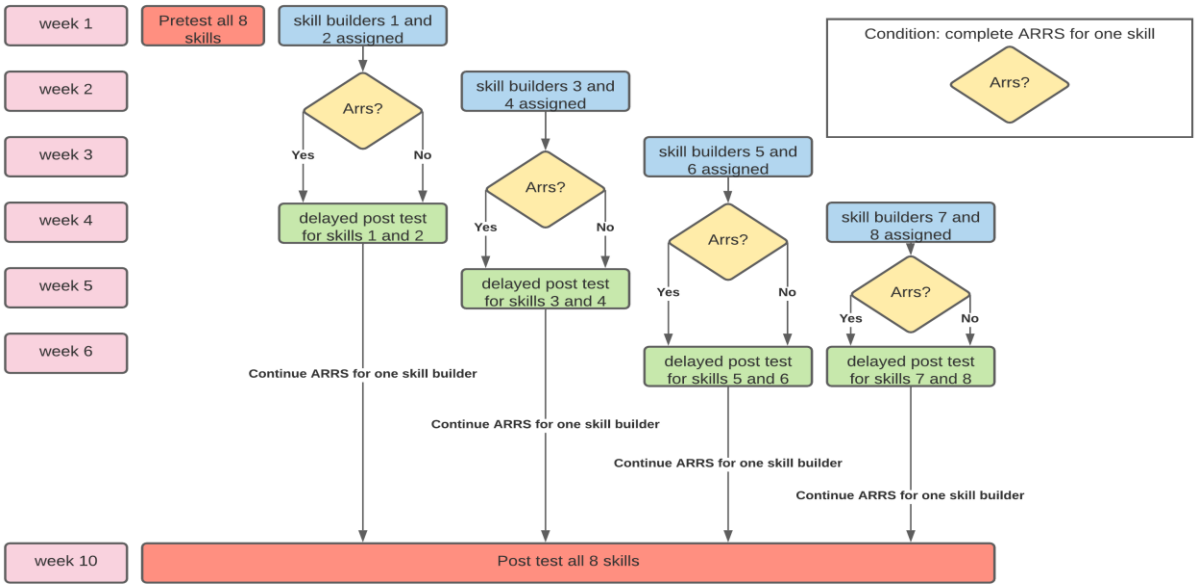

Figure 2: ARRS study

Skill Builder Number	Name of Skill Builder
1	Similar Triangles
2	Sum of Interior Angles in a Triangle
3	Pythagorean Theorem on the Coordinate plane
4	Trigonometric Ratios
5	Angles of Parallelograms
6	Area of Parallelograms
7	Area of Triangles
8	Area of Trapezoids

Table 1: ARRS skill builder Subjects

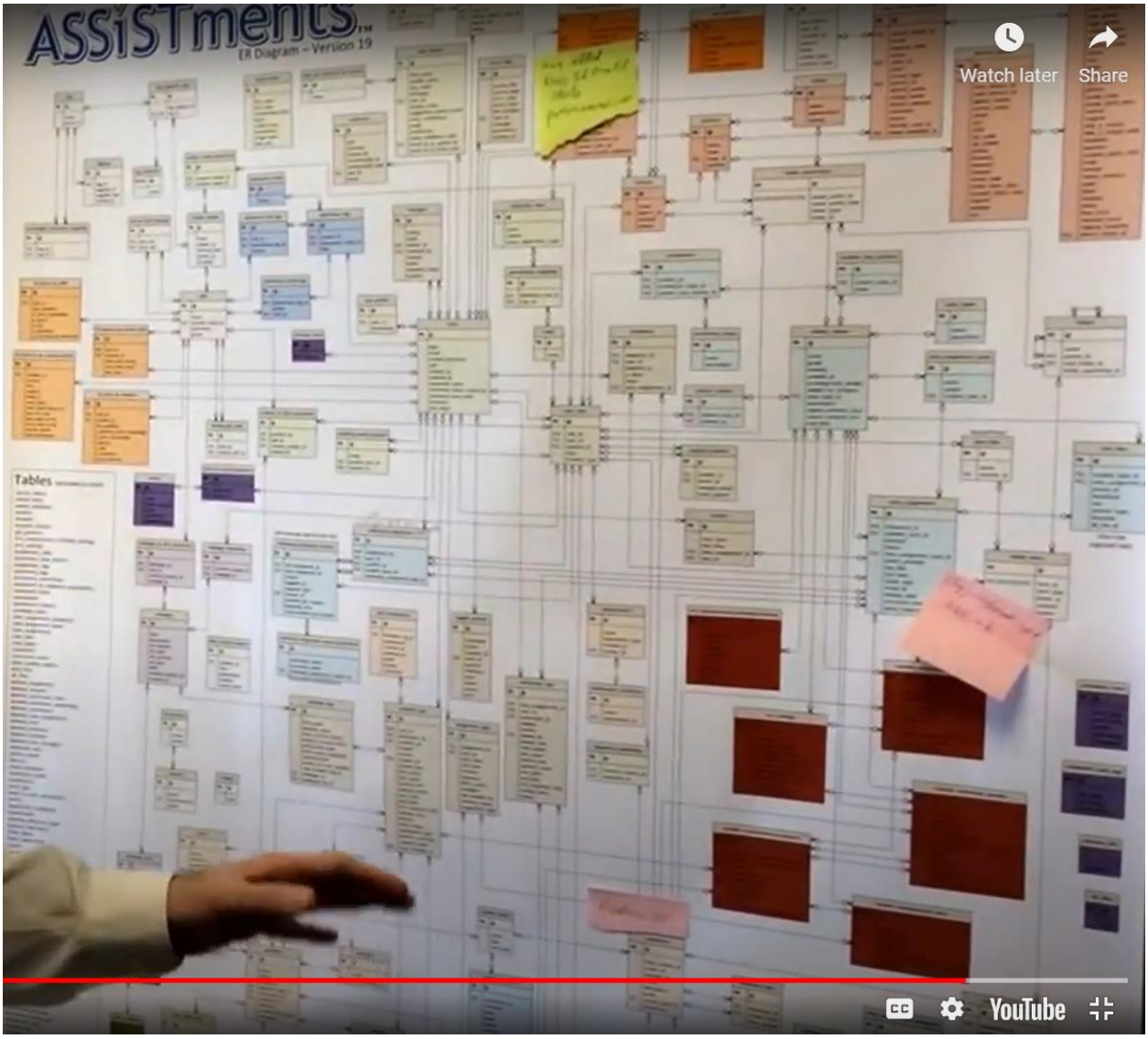

Figure 3: old ASSISTments ER diagram

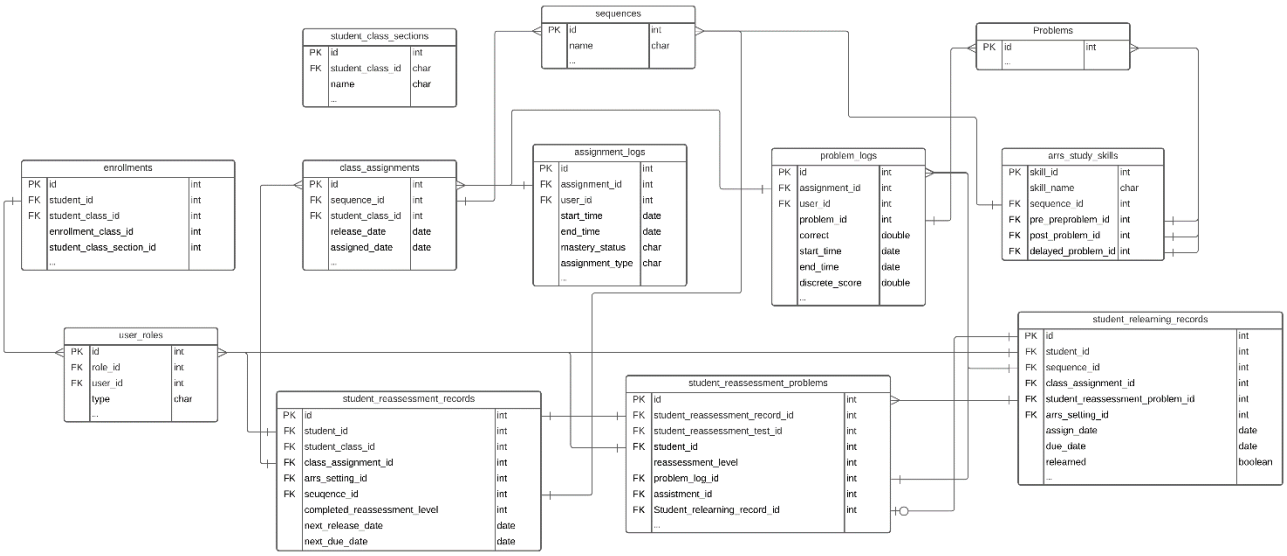

Figure 4: relevant ASSISTments ER diagram

Queries

Query 1:

```
DROP TABlE IF EXISTS public.arrs_study_skills;
Create Table public.arrs_study_skills (
        skill_id int,
        skill_name varchar(50),
        sequence_id int,
        pre_problem_id int,
        post_problem_id int,
        delayed_problem_id int,
        Primary Key (skill_id)
);
insert into public.arrs_study_skills (skill_id, skill_name, sequence_id, pre_problem_id, post_problem_id,
                                      delayed_problem_id) VALUES
(1, 'Similar Triangles', 6060, 71431, 71462, 71462),
(2, 'Sum of Interior Angles in a Triangle', 21257, 253976, 253888, 254051),
(3, 'Pythagorean Theorem on the Coordinate Plane', 459534, 1123592, 1123576, 1123600),
(4, 'Trigonometric Ratios', 469788, 837493, 837415, 837425),
(7, 'Area of Triangles', 487281, 171495, 171529, 171529),
(8, 'Area of Trapezoids', 10765, 1112877, 1112853, 1112865);
```

Query 2:

```
select * from public.arrs_study_skills
Select name, student_groups.student_id, student_groups.user_id, tests.skill_id, skill_name,
pre_problem_id, pre_test_score,  delayed_problem_id, delayed_test_score,
post_problem_id, post_test_score, num_problems as problems_completed_in_skill_builder,
mastery_status, avg_reassessment_score, reassessments_completed,
case when highest_level IS NOT NULL then highest_level else completed_reassessment_level end as
completed_reassessment_level,
relearings_attempted, relearnings_completed, arrs,
student_reassessment_record_id
FROM
(select name, enrollments.student_id, ur.user_id from
(select * from student_class_sections where student_class_id = 54176) as scs
inner join enrollments
on enrollments.student_class_section_id = scs.id
inner join (select * from user_roles where type = 'Student') as ur
on ur.id = enrollments.student_id) as student_groups

---------------------------------------------------------------------------------------------------------------------
left join
(SELECT pre_test.skill_id, skill_name, pre_test.user_id, pre_test_score,
 pre_problem_id, delayed_test_score, delayed_problem_id, post_test_score,
 post_problem_id
FROM
(SELECT skill_id, skill_name, pre_test_problems.user_id as user_id,
 correct as pre_test_score, problem_id as pre_problem_id
 from
 sequences
 inner join
 (select * from class_assignments where id = 2183073) as pre_test
 on pre_test.sequence_id = sequences.id
Inner Join
 (select * from problem_logs
 where problem_id in (71431, 253976, 1123592, 837493, 171495, 1112877)) as pre_test_problems
 on pre_test.id = pre_test_problems.assignment_id
 Inner Join arrs_study_skills on arrs_study_skills.pre_problem_id = pre_test_problems.problem_id) as
pre_test
 inner join
 (SELECT skill_id, delayed_test_problems.user_id as user_id,
  correct as delayed_test_score, problem_id as delayed_problem_id
 from
 sequences
 inner join
 (select * from class_assignments where id in(2197071, 2198924, 2198933)) as delayed_test
```

```
 on delayed_test.sequence_id = sequences.id
 Inner Join
 (select * from problem_logs
 where problem_id in (71462, 254051, 1123600, 837425, 171529, 1112865)) as delayed_test_problems
 on delayed_test.id = delayed_test_problems.assignment_id
 Inner Join arrs_study_skills
 on arrs_study_skills.delayed_problem_id = delayed_test_problems.problem_id) as delayed_test
 on delayed_test.user_id = pre_test.user_id and delayed_test.skill_id = pre_test.skill_id
  inner join
 (SELECT skill_id, post_test_problems.user_id as user_id, correct as post_test_score,
  problem_id as post_problem_id
 from
 sequences
 inner join
 (select * from class_assignments where id = 2232475) as post_test
 on post_test.sequence_id = sequences.id
 Inner Join
 (select * from problem_logs
 where problem_id in (71462, 253888, 1123576, 837415, 171529, 1112853)) as post_test_problems
 on post_test.id = post_test_problems.assignment_id
 Inner Join arrs_study_skills
 on arrs_study_skills.post_problem_id = post_test_problems.problem_id) as post_test
 on post_test.user_id = pre_test.user_id and post_test.skill_id = pre_test.skill_id) as tests
on tests.user_id = student_groups.user_id

-----------------------------------------------------------------------------------------------------------------------
left join
(select arrs_study_skills.skill_id, srr.id, srr.student_id, srr.sequence_id, avg_reassessment_score,
 reassessments_completed, completed_reassessment_level, student_reassessment_record_id,
 relearings_attempted, relearnings_completed, 1 as arrs, highest_level
 from
(select * from student_reassessment_records
 where sequence_id in (6060, 21257, 459534,469788,487281,10765)
 and class_assignment_id in (2181029,2181034,2181037,2181050,2181229,2181232)) as srr
 left join
(select student_reassessment_record_id, AVG(problem_logs.correct) as avg_reassessment_score,
 count(problem_logs.id) as reassessments_completed,
 max(case when problem_logs.correct = 1 THEN reassessment_level ELSE 0 END) as highest_level,
 count(relearning_aslogs.end_time) as relearnings_completed,
 count(relearning_aslogs.start_time) as relearings_attempted
 from
 (student_reassessment_problems
  left join problem_logs
  on problem_logs.id = student_reassessment_problems.problem_log_id
```

```
 left join student_relearning_records as srlr
 on student_reassessment_problems.id = srlr.student_reassessment_problem_id
left join assignment_logs as relearning_aslogs
 on relearning_aslogs.assignment_id = srlr.relearning_class_assignment_id
)
 Group by student_reassessment_record_id
) as srp
on srp.student_reassessment_record_id = srr.id
inner join arrs_study_skills
on arrs_study_skills.sequence_id = srr.sequence_id) as arrs
on arrs.skill_id = tests.skill_id and arrs.student_id = student_groups.student_id

-----------------------------------------------------------------------------------------------------------------
left join
(select enrollments.student_id, arrs_study_skills.skill_id, assignment_logs.start_time,
assignment_logs.end_time,
mastery_status, p_count.num_problems as num_problems
from
(select * from student_class_sections where student_class_id = 54176) as scs
inner join enrollments on enrollments.student_class_section_id = scs.id
inner join (select * from user_roles where type = 'Student') as ur on ur.id = enrollments.student_id
inner join
(select * from student_reassessment_records where sequence_id in (6060, 21257,
459534,469788,487281,10765)
and class_assignment_id in (2181029,2181034,2181037,2181050,2181229,2181232)) as srr
on enrollments.student_id = srr.student_id and srr.student_class_id = scs.student_class_id
left join assignment_logs
on assignment_logs.assignment_id = srr.class_assignment_id
and assignment_logs.user_id = ur.user_id
left join
(select assignment_id, user_id, count(id) as num_problems
 from problem_logs group by assignment_id, user_id) as p_count
on p_count.assignment_id = assignment_logs.assignment_id
and p_count.user_id = assignment_logs.user_id
inner join arrs_study_skills
on arrs_study_skills.sequence_id = srr.sequence_id
union
select enrollments.student_id, arrs_study_skills.skill_id, assignment_logs.start_time,
assignment_logs.end_time, mastery_status, num_problems
from
(select * from class_assignments where student_class_id = 54176
and sequence_id in (6060, 21257, 459534,469788,487281,10765)
and assignment_type_id =1) as skill_builders
left join assignment_logs
```

```
on assignment_logs.assignment_id = skill_builders.id
left join
(select assignment_id, user_id, count(id) as num_problems
 from problem_logs group by assignment_id, user_id) as p_count
on p_count.assignment_id = assignment_logs.assignment_id
and p_count.user_id = assignment_logs.user_id
left join (select * from user_roles where type = 'Student') as ur
on ur.user_id = assignment_logs.user_id
inner join enrollments
on enrollments.student_id = ur.id
inner join
(select * from student_class_sections where student_class_id = 54176) as scs
on enrollments.student_class_section_id = scs.id
inner join arrs_study_skills
on arrs_study_skills.sequence_id = skill_builders.sequence_id) as sbs
on sbs.student_id = student_groups.student_id and sbs.skill_id = tests.skill_id
```

R Code

Data loading and cleaning

```
install.packages("estimatr")
library(dplyr)
library(estimatr)

final_arrs_data <- read.csv(file='C:/Users/Jack/Documents/WPI/Thesis/ARRS study
results/final_arrs_data.csv')


int_arrs_data <- transform(final_arrs_data, arrs=as.numeric(arrs),
                           skill_id=as.numeric(skill_id),
                           problems_completed_in_skill_builder=
                             as.numeric(problems_completed_in_skill_builder),
                           post_test_score=as.numeric(post_test_score),
                           pre_test_score=as.numeric(pre_test_score),
                           delayed_test_score=as.numeric(delayed_test_score),
                           avg_reassessment_score=
                             as.numeric(avg_reassessment_score),
                           reassessments_completed=
                             as.numeric(reassessments_completed),
                           completed_reassessment_level=
                             as.numeric(completed_reassessment_level),
                           relearnings_completed=
                             as.numeric(relearnings_completed))

int_arrs_data$arrs[is.na(int_arrs_data$arrs)] <- 0
int_arrs_data$reassessments_completed[
  is.na(int_arrs_data$reassessments_completed)] <- 0
int_arrs_data$pre_test_score[is.na(int_arrs_data$pre_test_score)] <- 0
int_arrs_data$delayed_test_score[is.na(int_arrs_data$delayed_test_score)] <- 0
int_arrs_data$avg_reassessment_score[
  is.na(int_arrs_data$avg_reassessment_score)] <- 0
int_arrs_data$problems_completed_in_skill_builder[
  is.na(int_arrs_data$pre_test_score)] <- 0
int_arrs_data$relearnings_completed[
  is.na(int_arrs_data$relearnings_completed)] <- 0
int_arrs_data$completed_reassessment_level[
  is.na(int_arrs_data$completed_reassessment_level)] <- 0
int_arrs_data$mastery_status[is.na(int_arrs_data$mastery_status)] <-"incomplete"

clean_data <- int_arrs_data%>%group_by(student_id)%>%filter(sum(arrs)==3)
clean_data <- clean_data%>%group_by(student_id)%>%filter(mean(arrs)==.5)
```

Data analysis

```
clean_mod <- lm_robust(delayed_test_score~as.factor(skill_id)+
                         as.factor(student_id)+arrs+pre_test_score,
                       data=clean_data)
three_or_more_mod <- lm_robust(delayed_test_score~as.factor(skill_id)+
                           as.factor(student_id)+arrs+pre_test_score
                         , data=two_or_more)
summary(clean_mod)
summary(three_or_more_mod)


clean_mod <- lm_robust(post_test_score~as.factor(skill_id)+as.factor(student_id)
                       +arrs+pre_test_score, data=clean_data)
three_or_more_mod <- lm_robust(post_test_score~as.factor(skill_id)+
                           as.factor(student_id)+arrs+pre_test_score,
                         data=three_or_more)

summary(clean_mod)
summary(three_or_more_mod)
```