Exploring Prerequisite Relationships between Mathematical Concepts in Intelligent Tutoring Systems

Elaine Xu, Smeet Poladia, Zhou Lu

1 Abstract

Intelligent Tutoring Systems (ITSs) are an educational technology that provides students with a virtual learning environment, where every action of a student is tracked and recorded as a transaction. In MATHia, an ITS for school mathematics developed by Carnegie Learning, students are presented with workspaces, i.e, groups of problems on the same topic. Each problem further has many steps. Whenever a student solves a step correctly, they may learn a Knowledge Component (KC). For KCs within selected workspaces on granular data, we have used Gaussian Graphical Models (GGMs) to identify strongly related KCs, where the student performance metric for calculating correlations is success rate on initial opportunities of KC. We have implemented generalized linear mix-effects model to better understand and quantify the relationships between KCs. For workspaces on transaction data for an entire academic year, we identify strongly correlated workspaces using GGMs. The student performance metrics that were used are: success rate on first attempts across workspace, and assistance score. Then, we retrospectively study the effect of attempting the 'prerequisite' workspace on student success rate on first attempts across a workspace. We test the significant of this effect by using independent sample t-test, assuming unequal variances.

2 Introduction

Intelligent Tutoring Systems (ITSs) are an educational technology that provides students with a virtual learning environment and logs data of students' learning experience. Every action a student makes - from answering a question to requesting hints - is tracked by the system and recorded as a transaction. MATHia is an ITS for learning mathematics, developed by Carnegie Learning, that personalizes instruction for middle school and high school students. The client of this project is Dr. Vincent Aleven, who is the co-founder of Carnegie Learning and a professor at Human-Computer Interaction Institute, Carnegie Mellon University.

The project aims at making ITS more effective by detecting whether prerequisite relations among math topics can be detected in log data. If we can determine that mastering a certain math skill is necessary or can facilitate the learning of another skill, then the tutor system could train students on this prerequisite skill first and yield a better learning experience.

Some specific questions to be addressed in the paper are:

- How do we determine whether two math skills are related?
- What metrics and at what granularity of the metric should we use for evaluating learning and performance?
- How do we test whether workspace/skill A is prerequisite for B?

3 Data

The data for this study are provided by Carnegie Learning. Four datasets are used in this study. The first three datasets provide information about a random sample of students' performance on three different

workspaces in 2019-2020 academic year. The three workspaces are A: 'Analyzing Models of Two-Step Linear Relationships', B: 'Modeling Two-Step Expressions', and C: 'Using Scale Factor'. In this study, we assume that the knowledge components in workspace A are prerequisite for knowledge components in workspace B. Moreover, we believe that the workspace C should not be a prerequisite of two workspaces A and B, and comes prior to them in the curriculum. The reader should refer to Corbett et al. (2000) for details about this prerequisite relationship. The fourth dataset is a larger dataset which contains a random sample of 500 students' performance in all Course 2 (Grade 7) MATHia workspaces in the 2019-2020 academic year.

Table 1 provides some general information about the number of unique students, the number of unique knowledge components, and the number of unique steps (opportunities) in each data set.

In Table 2, we provide descriptions of selected variables. Within each dataset, there are multiple workspaces. Within each workspace, there are multiple problems. Within each problem, multiple steps (opportunities) are presented to students. A unique knowledge component is mapped to each step. Readers can refer to Fancsali et al. (2021) for more details about the variable descriptions.

In this study, we work on a filtered MATHia Course 2 dataset, which includes a "shipped" curriculum sequence, to focus on a smaller set of workspaces. The smaller MATHia Course 2 dataset contains 500 students, 223 unique knowledge components, and 65685 unique steps (opportunities).

Later in the study, a smaller dataset, which contains only 500 students for each workspace A, B and C, is used to quantify the influence of knowledge components. Since the opportunity column which allows us to quantify the influence only exists in a side dataset which includes 500 randomly selected students, we used this smaller dataset instead of the complete ones in Table 1.

Dataset	**# of Students**	**# of Unique Knowledge Components**	**# of Unique Steps (Opportunities)**
A: Analyzing Models of Two-Step Linear Relationships	29949	7	7
B: Modeling Two-Step Expressions	27005	9	9
C: Using Scale Factor	19521	4	29
MATHia Course 2	500	964	117210

Table 1: General Overview of Datasets

Variable Name	Values	Description
Anno.Student.Id	Integer	anonymous student identifier
Time	Timestamp	Timestamp in UNIX epoch time
Problem.Name	Character	Identifier for the problem
Step.Name	Character	Identifier for the problem-step
Action	"Attempt","Done", "Hint Request", "Hint Level Change"	Student's action for the problem-step. "Attempt" = student made a problem-solving attempt, "Done" = student clicked the "Done" button required to complete a problem, "Hint Request" = Student requests a hint, "Hint Level Change" = Student requests a hint at a "deeper" level
Outcome	"OK", "ERROR", "BUG", "INITIAL_HINT", "HINT_LEVEL_CHANGE"	"OK" = correct, "ERROR" = error that isn't specifically tracked for JIT feedback, "BUG" = error that is tracked for just-in-time, context-sensitive feedback, "INITIAL_HINT" = first-level hint is provided, "HINT_LEVEL_CHANGE" = a "deeper" level hint is provided
KC..MATHia.	Character	The skill or knowledge component (KC) tracked by MATHia for this problem-step
CF (Skill New p-Known)	Real Number	Bayesian Knowledge Tracing skill estimate after this action (i.e., semantic event)

Table 2: Descriptions of Variables in Datasets

4 Methods

4.1 Prerequisite relations between KCs from workspaces A, B and C

4.1.1 Gaussian Graphical Models (GGMs)

A Gaussian Graphical Model (GGM) (Bhushan et al., 2019) is an exploratory analysis tool that provides an easy to grasp overview of relationships between knowledge components (KCs) from workspaces A, B and C. A GGM has KCs, as nodes, and edges (that connect those nodes) which visualize the relationship between the KCs. The thickness of these edges represents the strength of relationships. The edge is green in color if the correlation is positive and red otherwise. KCs which are strongly correlated are placed spatially close to each other in the plot.

To obtain a GGM, we need to have a correlation matrix as an input. We calculate correlation between KCs where each observation in the data represents the logit transformation of a student's success rate on initial opportunities of KCs. The logit transformation is used because the underlying data is assumed to have a multivariate normal distribution, and the logit transforms the success rate from a 0 to 1 scale to an $-\infty$ to ∞ scale to match the normal distribution. More information about initial opportunities is provided in section 4.1.2 of this paper. We observe success rate of 0 and 1 in our data, which are respectively converted to 0.0001 and 0.9999. Here, the correlation is calculated using the full information maximum likelihood

procedure (FIML). Correlations obtained through this procedure are able to handle missing data, robust to deviations from multivariate normality and are less biased estimates (Bhushan et al., 2019).

In GGM, the thickness of the edges between various KCs (depicted by nodes) represents the strength of relationships and are interpreted as partial correlation coefficients. Since relationships estimated by GGM are interpreted as partial correlation coefficients, we reduce the risk of finding any spurious relations between any two KCs, that is, caused by a third KC in the data. These partial correlations are estimated by the GGM using the glasso algorithm (https://cran.r-project.org/web/packages/glasso/index.html). The glasso algorithm also helps us obtain a sparse graph where only important partial correlation coefficients are represented and unimportant partial correlation coefficients are forced to zero.

4.1.2 Initial Opportunities

To better understand students' performance on each knowledge component, for each student, we used initial opportunities instead of all opportunities to evaluate that student's understanding of mathematical concepts. For each knowledge component, students are given multiple opportunities (steps) until students demonstrate mastery of that mathematical skill. Different numbers of opportunities are given to each student based on their performance. In order to prevent smoothing out differences among students, we chose to use initial opportunities which are better indications of students' mastery level.

We utilized Gaussian Graphical Models to determine the cutoff point for initial opportunities. We generated multiple Gaussian Graphical Models using different cutoff points. For easier visual examination, lines with partial correlations larger than 0.05 in absolute value are shown in the graph. The number of opportunities which produces the model with the best structure is selected as our final cutoff point for initial opportunities. The model with the best structure is the one that gives us the strongest partial correlations between workspaces.

4.1.3 Generalized Linear Mixed-Effects Model

To better understand the partial correlations between two KCs from the Gaussian Graphical Model and to quantify the influence one KC has on another, we adopted a mixed effects logistic regression approach. If there exist some prerequisite relationships between two knowledge components KC1 and KC2, and we assume KC2 is a prerequisite of KC1, then student who knows KC2 should have better performance on KC1 than student who does not know KC2. At the same time, whether a student knows KC1 should not have much effect on student's performance on KC2.

The tutor system calculates a score between 0 and 1 to indicate a student's grasp of a certain KC (stored in the variable CF.Skill.New.P.Known; see Table 2), and deems a student has mastered this KC if the score is above 0.95. We calculated an indicator variable of whether a student has mastered KC2 using this score as one of the independent variable. Other predictors include KC 1 opportunity (the number of times a student has encountered KC 1) and an interaction term. Student ID is the random effect term since its variability cannot be explained by the predictors of the model. The performance metric used is whether student correctly answered the step on their first attempt. For this analysis, we used the smaller datasets that include 500 students since the opportunity column only exists in the small datasets.

The final model looks like:

$$glmer(First.Attempts.KC1 \sim 1 + KC1Opportunity + know.KC2 + KC1Opportunity * know.KC2 + (1|Student.ID))$$

4.2 Prerequisite relations between workspaces from MATHia Course 2

4.2.1 Gaussian Graphical Models (GGMs)

We use GGMs to identify relationships between workspaces from student data of MATHia Course 2. The interpretation from GGM having workspaces remains the same as that of the GGM having KCs. A GGM

has workspaces, as nodes, and edges (that connect those nodes) which visualize the relationship between them. The thickness of these edges represents the strength of relationships and are interpreted as partial correlation coeffients. The edge is green in color if the correlation is positive and red otherwise. Workspaces which are strongly correlated are placed spatially close to each other in the plot.

However, few things that have changed:

- We use 76 workspaces that are a part of the MATHia Couse 2 shipped curriculum.
- Two metrics for student performance are used:
  - Success rate of first attempts of all steps across a workspace. We apply the logit transofrmation to this data as it transforms the success rate from a 0 to 1 scale to an $-\infty$ to ∞ scale to match the normal distribution. When success rate of 0 and 1 is observed, the values are respectively converted to 0.0001 and 0.9999.
  - Assistance score across a workspace. We apply a log transformation to this data transforms the assistance score from a 0 to ∞ scale to an $-\infty$ to ∞ scale to match the normal distribution. When an assistance score of 0 is observed, we use 0.0001 instead.
- In the data, not all students have attempted all workspaces. Any given workspace is attempted by at most 210 students out of 500 that we have. We are unable to calculate correlation matrices for both of the above mentioned metrics using the full information maximum likelihood procedure because there is too much casewise missing data. Instead, we compute pairwise correlations between workspaces and convert the resulting correlation matrix to positive definite using a smoothing technique. The correlation matrix thus obtained is used as an input for the GGM.

4.2.2 Retrospective study between related workspaces

From the GGMs, we obtain the pairs of related workspaces and their partial correlation coefficients. For any given pair of related workspaces, we consider a retrospective study design and test whether the performance of students who have first attempted the prerequisite workspace is greater than that of students who haven't attempted the prerequisite workspace first.

The procedure we follow:

- Consider X and Y being two related workspaces from the GGM having some partial correlation coefficient. We test whether X is a prerequisite of Y.
- We compute the performance of two groups of students. Students in one group have attempted workspace X before attempting workspace Y, and students in the other group have not attempted workspace X before attempting workspace Y.
- For every student is any of the two groups, the student performance is the mean success rate across first attempt for all steps of workspace Y.
- Now, we have:
  - μ_X: Mean student performance on workspace Y for students who have attempted workspace X before attempting workspace Y.
  - μ_Y: Mean student performance on workspace Y for students who have not attempted workspace X before attempting workspace Y.
- We use independent sample t-test, assuming unequal variances, to test our hypothesis. In this case, we have $H_0 \ : \ \mu_X \leq \mu_Y$ versus $H_1 \ : \ \mu_X > \mu_Y$.
  - If we reject H_0, the null hypothesis, we can say that workspace X is a prerequisite for workspace Y because student performance is significantly greater for students who have attempted the prerequisite than those who haven't.

– If we do not reject H_0, the null hypothesis, we say that X is not a prerequisite of Y.

- Now, we repeat the above steps and test whether Y is a prerequisite of X.

The procedure mentioned above is repeated for all the related workspaces we have obtained from the GGM.

5 Results

5.1 Prerequisite relations among KCs from workspaces A, B and C

5.1.1 Gaussian Graphical Models (GGMs)

After comparison, the initial 2 opportunities gave us the best structured GGM (Figure 1). Therefore, for the rest of the analysis, we are going to use the initial 2 opportunities to evaluate students' performance. More details about the GGMs for different numbers of opportunities are available in page 7, Appendix 2. The highest partial correlation is 0.1917 and is observed between 'find y, any form-1' and 'identifying units-1', having nodes 4 and 5, respectively. To make sure GGM is interpretable, only partial correlation having absolute value of 0.05 or greater are shown. The GGM has found 165 pairs of KCs which have important partial correlation coefficients.

In Table 3, we explore the top 10 strongest relationships between KCs where the student performance metric is success rate on initial 2 opportunities of a KC. From Table 3 and Figure 1, we notice that KCs cluster within the same workspace. This might be because they are being learned by students at the same time.

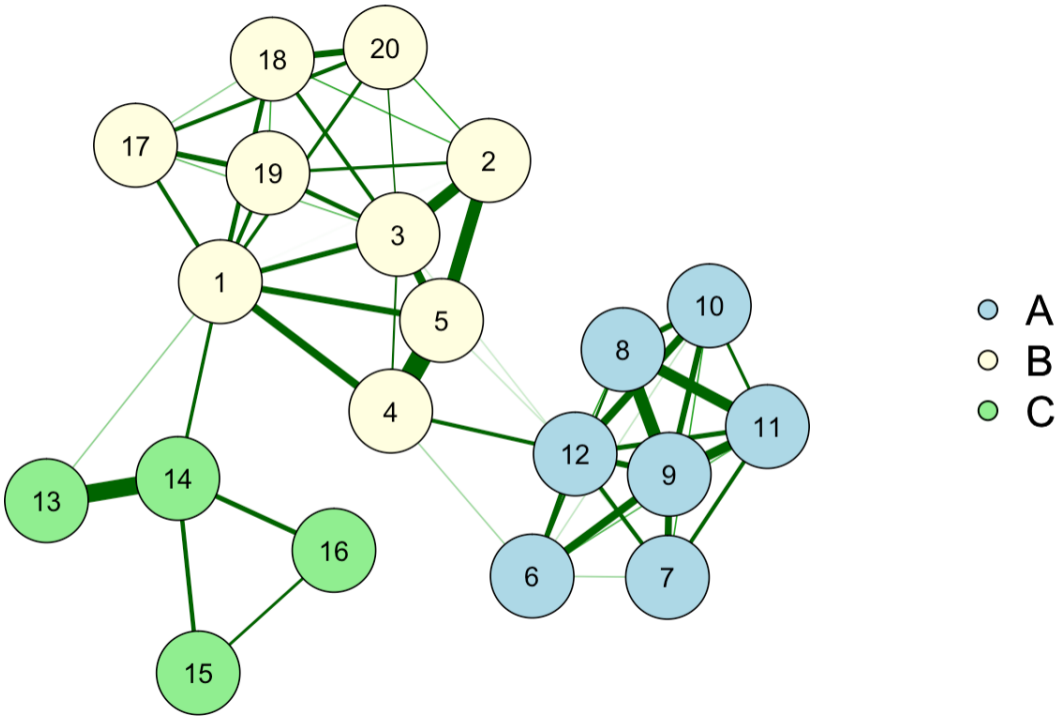


Figure 1: GGM showing relationships between KCsin workspaces: A, B and C. A: Analyzing Models of Two-Step Linear Relationships, B: Modeling Two-Step Expressions, and C: Using Scale Factor.

Node (1)	KC (1)	Node (2)	KC (2)	Partial Correlation
4	find y, any form-1	5	identifying units-1	0.1917
13	scale drawings-3 determine unknown measure, complex scale factor	14	scale drawings-3 determine unknown measure, simple scale factor	0.1703
8	match dep expression with description	9	match indep expression with description	0.1376
2	enter given, reading numerals-1	5	identifying units-1	0.1363
2	enter given, reading numerals-1	3	enter given, reading words-1	0.1284
8	match dep expression with description	11	match linear-term expression with description	0.1274
9	match indep expression with description	11	match linear-term expression with description	0.1164
3	enter given, reading words-1	5	identifying units-1	0.1063
1	define variable-1	4	find y, any form-1	0.1052
10	match intercept expression with description	12	match slope expression with description	0.1046

Table 3: Top 10 Strongest Relationships between KCs

5.1.2 Generalized Linear Mixed-Effects Model

A generalized linear mixed-effects model (glmer) was applied to different pairs of knowledge components, and the model results were recorded in Table 5. Main effect indicates how much knowing KC2 would influence student's likelihood of mastering KC1. The opportunity column shows the coefficient of the number of times a student has encountered KC1. The interaction term tells us if knowing KC2 would make learning KC1 faster. The coefficients with an asterisk are not statistically significant (p-value > 0.05).

In the table, the opportunity column has all positive values, which makes sense since it means the more times a student encounters a KC, the more likely they are to master it. The results also align with what we saw in the Gaussian Graphical Models. For example, there is no line connecting KC1 and KC18 in Figure 3, and the coefficients of main effect in Table 4 for them are not significant. It is also surprising to note that the coefficients for the interaction term are all negative, which means learning KC2 would have a negative effect on how fast students learn KC1. One explanation is: The coefficients for opportunity and interaction being relatively the same size, together with the fact that knowing the presumably prerequisite KC brings up the probability of answering "result" KC correct to nearly 1, causes the interaction coefficient to be close to 0 since there is no room for performance improvement.

The R code for this procedure can be found in Appendix 3.

"Prereq" KC	KC	Main Effect	Opportunity	Interaction
1 (define variable)	5 (identifying units)	1.21	0.18	-0.13
5 (identifying units)	1 (define variable)	2.43	0.34	-0.31
4 (find y, any form-1)	5 (identifying units)	1.20	0.10	-0.07
5 (identifying units)	4 (find y, any form-1)	0.50	0.08	-0.04
18 (write expression, negative slope-1)	1 (define variable)	1.45*	0.63	-0.60
1 (define variable)	18 (write expression, negative slope-1)	30.98*	28.63*	-28.90*
18 (write expression, negative slope-1)	5 (identifying units)	1.20	0.12	-0.08
5 (identifying units)	18 (write expression, negative slope-1)	4.00	1.99	-1.64

Table 4: Glmer Results (starred coefficients are **not** significant)

5.2 Prerequisite relations among workspaces from MATHia Course 2

5.2.1 Gaussian Graphical Models (GGMs)

Figure 2 and Table 5 show the GGM and the top 10 strongest relationships between workspaces, when the metric for student performance is success rate of first attempts of all steps across a workspace. The highest partial correlation is 0.4122 and is observed between 'solving simple percent problems' and 'using proportion to solve percent problems', having nodes 61 and 74, respectively (located in the top-center region of the plot). To make sure GGM is interpretable, only partial correlation having absolute value of 0.1 or greater are shown. The GGM has found 661 pairs of workspaces which have important partial correlation coefficients.

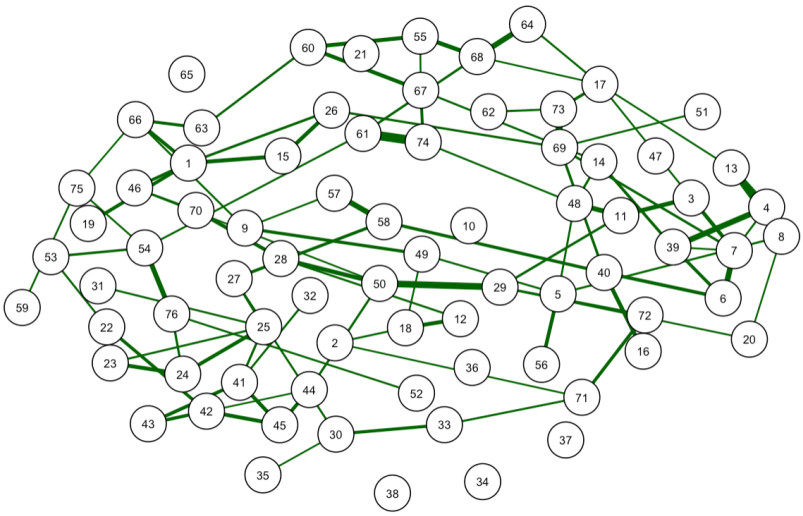


Figure 2: GGM showing relationships between workspaces. Student performance metric is success rate for first attempts of all steps across the workspace.

Node (1)	Workspace (1)	Node (2)	Workspace (2)	Partial Correlation
61	solving simple percent problems	74	using proportions to solve percent problems	0.4122
69	graphs of equations	73	using graphs to solve equations	0.3293
29	scale drawings 3	50	volume surface area right prism vol-backward	0.3112
8	comparing theoretical and experimental probabilities	13	determining probabilities	0.2875
4	calculating compound probabilities	13	determining probabilities	0.2743
4	calculating compound probabilities	39	simulating compound events	0.2702
6	comparing characteristics of data displays	7	comparing populations using data displays	0.2672
64	converting with fractional percents	68	fractional percent models	0.2587
57	ratio proportion change3	58	ratio proportion change4	0.2556
11	critical attributes of similar figures	48	using scale drawings	0.2357

Table 5: Top 10 Strongest Relationships between Workspaces when Metric is Success Rate

Figure 3 and Table 6 show the GGM and the top 10 strongest relationships between workspaces, when the metric for student performance is assistance score across a workspace. The highest partial correlation is 0.3527 and is observed between 'comparing characteristics of data displays' and 'comparing populations using data displays', having nodes 6 and 7, respectively (located in the top-right region of the plot). To make sure GGM is interpretable, only partial correlation having absolute value of 0.1 or greater are shown. The GGM has found 638 pairs of workspaces which have important partial correlation coefficients.

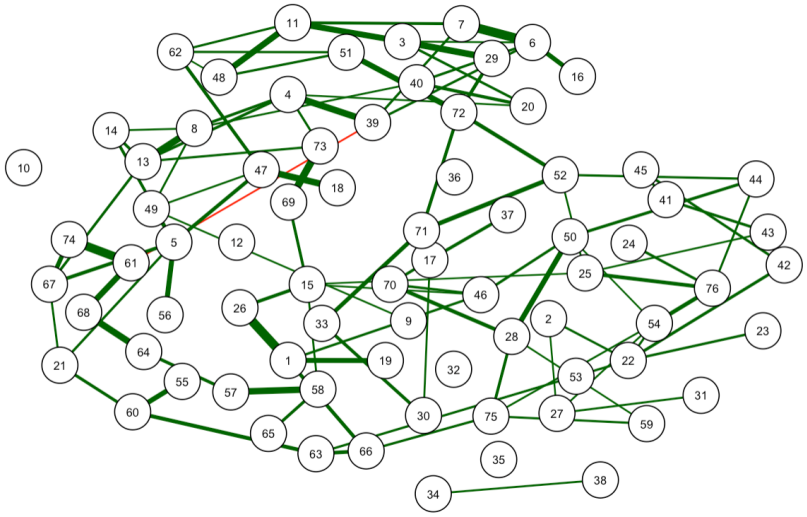


Figure 3: GGM showing relationships between workspaces. Student performance metric is assistance score across the workspace.

Node (1)	Workspace (1)	Node (2)	Workspace (2)	Partial Correlation
6	comparing characteristics of data displays	7	comparing populations using data displays	0.3527
1	adding and subtracting integers	26	multiplying and dividing integers	0.3512
61	solving simple percent problems	74	using proportions to solve percent problems	0.3369
69	graphs of equations	73	using graphs to solve equations	0.2926
8	comparing theoretical and experimental probabilities	13	determining probabilities	0.2747
4	calculating compound probabilities	39	simulating compound events	0.2677
64	converting with fractional percents	68	fractional percent models	0.2668
3	calculating angles	29	scale drawings 3	0.2646
18	identifying signs of starting values and rates	47	understanding volume of right prisms	0.2644
57	ratio proportion change3	58	ratio proportion change4	0.2534

Table 6: Top 10 Strongest Relationships between Workspaces when Metric is Assistance Score

We observe 7 pairs of workspaces that appeared in Table 5 as well as Table 6. These pairs of related workspaces include: 'comparing characteristics of data displays' and 'comparing populations using data displays', 'solving simple percent problems' and 'using proportions to solve percent problems', and 'graphs of equations' and 'using graphs to solve equations', among others. We also observe that partial correlation coefficients among the most strongly correlated pairs of workspaces are lower when student performance metric is assistance score than when the metric is success rate of first attempts across the workspace. In Figure 3, we also see an edge which is red in color, implying a partial correlation coefficient between 'simulating compound events' (39) and 'solving simple percent problems' (61) is less than -0.1. This might be because of the noise in the data and the fact that assistance score being a simple metric of student performance does not capture true student learning. More about this is mentioned in the Discussion section of the paper.

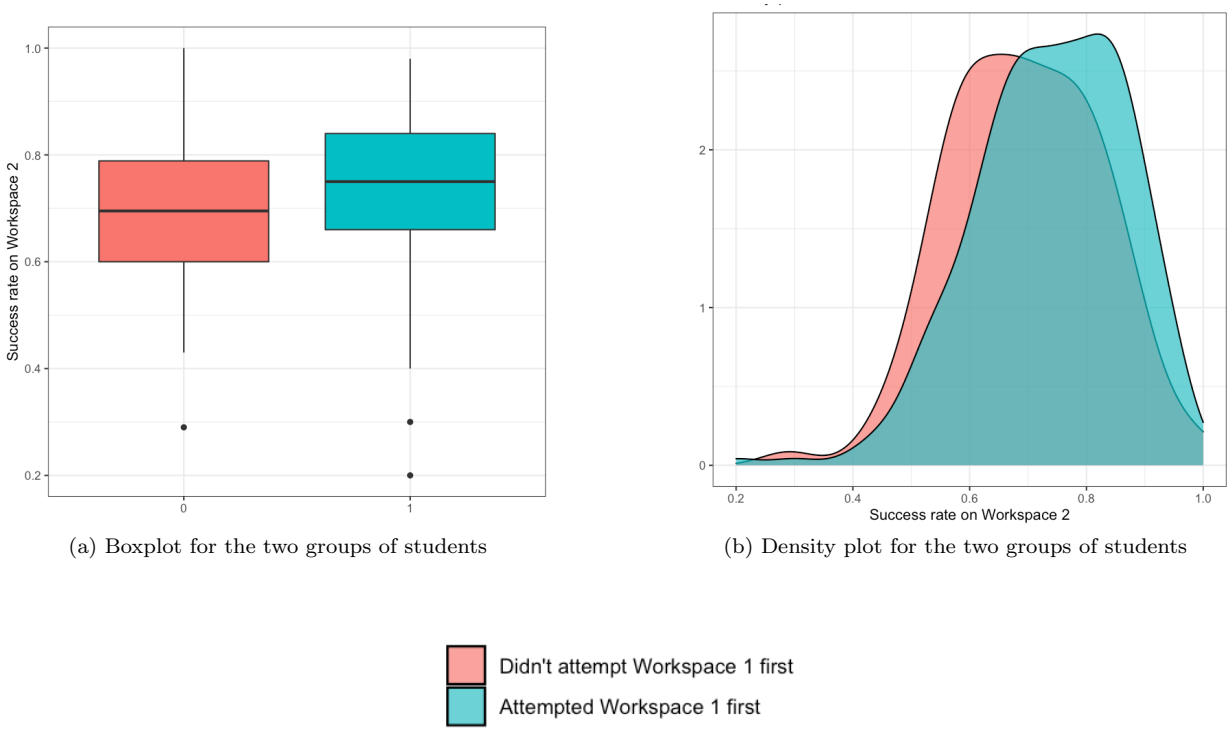
5.2.2 Retrospective study between related workspaces

For all the pair of workspaces obtained through both the GGMs, we consider a restrospective study design and test whether the performance of students after attempting the prerequisite workspace first, is statistically greater than performance of students who didn't attempt the prerequisite workspace first.

Consider Workspace 1: 'fractional percent models' and Workspace 2: 'solving simple percent problems', and we will show that Workspace 1 is a possible prerequisite of Workspace 2. From Figure 4, we can see that the median success rate on Workspace 2 is higher for students who attempted Workspace 1 first than for students who didn't attempt Workspace 1 first. The distribution for success rate on Workspace 2 appears to be shifted to right for students who attempted Workspace 1.

The mean success rate on Workspace 2 for students who attempted Workspace 1 first is 0.7388, and is 0.6944 for those who didn't attempt Workspace 1 first. From the data we have, 258 students attempted Workspace 1 first and 102 students didn't attempt Workspace 1 first. After performing independent sample t-test (assuming unequal variances) to check whether mean success rate on Workspace 2 for students who

attempted Workspace 1 first is greater than that of students who didn't attempt Workspace 1 first, we obtain a p-value of 0.0018. At 5% level of significance, we conclude that there is significant difference in mean success rate on Workspace 2 when students first attempt Workspace 1. Therefore, Workspace 1 is a possible prerequisite for Workspace 2.



(a) Boxplot for the two groups of students

(b) Density plot for the two groups of students

Figure 4: Distribution of success rate on Workspace 2 for students who have first attempted Workspace 1 (the prerequisite), and for students who haven't. Workspace 1: 'fractional percent models' and Workspace 2: 'solving simple percent problems'.

To check prerequisite relationship between other workspaces, the reader can check out the accompanying R Shiny application.

5.2.3 Workspaces with prerequisite relationships

Table 7 gives a few examples of pairs of workspaces which have a prerequisite relation. The pairs are obtained from GGMs in Figure 2 and Figure 3 and their prerequisite relationships are confirmed by retrospective study design. In the accompanying CSV files, the reader may find all the workspaces which have a prerequisite relationship.

"Prereq" Workspace	**Workspace**
comparing populations using data displays	graphs of equations
simplify order of ops expression numeric contrast addsub multdiv	simplify order of ops expression variable contrast addsub multdiv
adding and subtracting integers	ratio proportion change3
picture algebra mix variable	worksheet grapher a1 direct variation
worksheet grapher a1 solving 2step int	worksheet grapher a1 direct variation
linear relations 1	simplify order of ops expression numeric mix type complex

Table 7: Examples of Workspaces where Prerequisite Relations are Identified

6 Discussion

To determine the prerequisite relations between workspaces and between KCs, the metric of student performace we have used is either success rate or assistance rate. They are useful, but simple metrics and the may not capture the true student learning. As a next step, we would like to explore other metrics of student learning and then examine the prerequisite relation between workspaces or KCs, using methods mentioned in this paper.

To know strongly related KCs through the GGM, the metric of student performance used is success rate over initial opportunities. The number of initial opportunities which produces the best GGM is selected. Using this approach might tend to a GGM which overfits the data. As a next step, we would explore options of student performance metric where we don't have the issue of overfitting.

When exploring the relationships between KCs, even though the glmer method is a good way to quantify correlations between knowledge components, it does not infer causal relationships. Time order is a challenge here since it does not take into account the order of a student learning certain knowledge components. The order of topics is usually fixed, which makes it harder to test for prerequisites.

In the GGMs, where we examine relationships between workspaces, we observe that there are some workspaces where links are shown in red line, implying negative partial correlation coefficient. The reason for this might be attributed to the noise in the data. Speaking practically, high performance on one workspace cannot imply low performance on another workspace. We believe that as student performance metric is improved, we will no longer observe any important negative partial correlations between workspaces.

The partial correlation coefficients estimated by the glasso algorithm, range from as high as 0.45 in some of the GGMs to as low as 0.01. It is known that the glasso algorithm keeps only the important partial correlation coefficients and forces the unimportant ones to zero. Bhushan et al. (2019) demonstrated this using a non-parametric bootstrap method. However, this does not imply that the partial correlation coefficients are statistically different from zero. To test this, a separate test of significant is required which is not performed for our analysis in this paper. There do not yet exist widely available significance tests for lasso-penalized GGMs.

We have used retrospective study design when we want to identify prerequisite order between related workspaces. Since there is no randomization here, this type of study may be biased. Since we are dealing with data from 500 students only, for some pairs of related workspaces, we have found that data is

highly unbalanced. For instance, in one case there were 200 students who didn't attempted the 'prerequisite' workspace first, and only 4 students who have attempted it first. In such a case, results from independent sample t-test would not be reliable. To get more accurate and reliable results on the prerequisite relation between workspaces, Carnegie Learning could extend the idea of retrospective study design, mentioned in Section 4.2.2 of this paper to a dataset with more students. By doing this, the data for a pair of workspace might not be as unbalanced as when there are only have 500 students. Student performance metrics can also be extended from success rate and assistacne score to a linear combination of succes rate and assistance score, number of BUG outcomes, and weighted average success rate (where weights are assigned to outcomes - 'OK':5, 'BUG':4, 'HINT':3, 'HINT LEVEL CHANGE':2, 'ERROR':0).

This problem could be solved if we can design a randomized experiment to specifically determine prerequisite relations. We recommend Carnegie Learning to conduct randomized experiments between workspaces where we found a prerequisite relation and even between workspaces which show strong relationships in the GGMs. In these randomized experiments, students should be randomly assigned to treatment group (where they attempt the prerequisite workspace) and control group (where they do not attempt the prerequisite). The independent sample t-test should then give a more reliable idea about the difference in student performance between two groups.

Apart from conducting randomized experiments, we would recommend Carnegie Learning to use the information about prerequisite relations obtained with regard to KCs and workspaces, for improving students' learning performance and making MATHia more effective. Along with this paper, the accompanying CSV files give details about the prerequisite relationships between KCs and workspaces. Educational psychologists and specialists who create content for MATHia should be made aware of pairs of KCs/workspaces for which we found prerequisite relations using student log data. Using techniques in their discipline, it would be worth investigating the results of prerequisite relations that they have.

7 References

Bhushan, N.,, Mohnert, F., Sloot, D., Jans, L., Albers, C., and Steg, L. (2019), "Using a Gaussian Graphical Model to Explore Relationships Between Items and Variables in Environmental Psychology Research." *Frontiers in Psychology 10.*

Gauthier, G., Frasson, C., and VanLehn, L. (2000), "Learning Factors Analysis – A General Method for Cognitive Model Evaluation and Improvement." *Intelligent Tutoring Systems: 5th International Conference, ITS 2000*, Montreal, Canada, June 19-23, 2000: Proceedings. Berlin: Springer.

Fancsali, S.E. (2021), *Carnegie Learning MATHia 2019-2020 DataShop Documentation* Pittsburgh: Carnegie Learning.

Koedinger, K. R., Baker, R. S. J. d., Cunningham, K., Skogsholm, A., Leber, B., Stamper, J. (2010). A data repository for the EDM community: The PSLC datashop. In S. Ventura, C. Romero, M. Pechenizkiy, R. S. J. d. Baker (Eds.), *Handbook of educational data mining (pp. 43–55).* Boca Raton, FL: CRC Press.

R Core Team (2017), R: *A language and environment for statistical computing.* R Foundation for Statistical Computing, Vienna, Austria. http://www.r-project.org/index.html

RStudio Team (2020), *R Studio: Integrated Development Environment for R.* RStudio, PBC, Boston, MA. https://www.rstudio.com/

Appendix 1

In this technical appendix, we use data wrangling techniques to merge the data on the three workspaces and create a dataframe which has success rate for opportunities for each student per KC.

```
head(model_2_factor)
```

```
##                         Anon.Student.Id          Session.Id         Time
## 1 0001368e-a25a-4541-92b0-057c79ff2ebd no_session_tracking 1.579466e+12
## 2 0001368e-a25a-4541-92b0-057c79ff2ebd no_session_tracking 1.579466e+12
## 3 0001368e-a25a-4541-92b0-057c79ff2ebd no_session_tracking 1.579466e+12
## 4 0001368e-a25a-4541-92b0-057c79ff2ebd no_session_tracking 1.579466e+12
## 5 0001368e-a25a-4541-92b0-057c79ff2ebd no_session_tracking 1.579466e+12
## 6 0001368e-a25a-4541-92b0-057c79ff2ebd no_session_tracking 1.579466e+12
##                       Level..Workspace.Id.
## 1 worksheet_grapher_a1_patterns_2step_expr
## 2 worksheet_grapher_a1_patterns_2step_expr
## 3 worksheet_grapher_a1_patterns_2step_expr
## 4 worksheet_grapher_a1_patterns_2step_expr
## 5 worksheet_grapher_a1_patterns_2step_expr
## 6 worksheet_grapher_a1_patterns_2step_expr
##                                    Problem.Name  Step.Name Selection  Action
## 1 worksheet_grapher_a1_patterns_2step_expr_067 Unit-Indep           Attempt
## 2 worksheet_grapher_a1_patterns_2step_expr_067 Unit-Indep           Attempt
## 3 worksheet_grapher_a1_patterns_2step_expr_067   Q1-Indep           Attempt
## 4 worksheet_grapher_a1_patterns_2step_expr_067     Q1-Dep           Attempt
## 5 worksheet_grapher_a1_patterns_2step_expr_067   Q2-Indep           Attempt
## 6 worksheet_grapher_a1_patterns_2step_expr_067     Q2-Dep           Attempt
##                                                 Input Outcome Help.Level
## 1 {"escape-in-messages" : "false", "value" : "time"}     BUG          0
## 2 {"escape-in-messages" : "false", "value" : "week"}      OK          0
## 3    {"escape-in-messages" : "false", "value" : "1"}      OK          0
## 4   {"escape-in-messages" : "false", "value" : "19"}      OK          0
## 5    {"escape-in-messages" : "false", "value" : "2"}      OK          0
## 6   {"escape-in-messages" : "false", "value" : "18"}      OK          0
##   Attempt.At.Step               KC.Model.MATHia.           CF..Ruleid.
## 1               1             identifying units-1 label instead of unit
## 2               2             identifying units-1
## 3               1    enter given, reading words-1
## 4               1              find y, any form-1
## 5               1 enter given, reading numerals-1
## 6               1              find y, any form-1
##   CF..Etalon. CF..Skill.Previous.p.Known. CF..Skill.New.p.Known.
## 1       weeks                       0.25              0.2444444
## 2       weeks                         NA                     NA
## 3           1                       0.10              0.4600000
## 4          19                       0.10              0.4600000
## 5           2                       0.10              0.4600000
```

```
## 6          18                         0.46              0.8116279
##   CF..Workspace.Progress.Status.               CF..Semantic.Event.Id.
## 1                      GRADUATED c5bdbc70-fa4a-4d14-b9e2-4fc73b3955b4
## 2                      GRADUATED 3e656668-7625-4404-88c0-1ac9ad181de5
## 3                      GRADUATED 8efaefb7-0168-4687-b0d1-3447d7d34dc6
## 4                      GRADUATED 379dd6de-21ff-49f0-97ed-7f3a79c85641
## 5                      GRADUATED 2d589f2d-f37f-4093-a875-ac95c7d66616
## 6                      GRADUATED 6470b391-3603-4ecf-9e00-ad65a4e1f17f
```

```
head(analyze_2_factor)
```

```
##                         Anon.Student.Id          Session.Id         Time
## 1 0001368e-a25a-4541-92b0-057c79ff2ebd no_session_tracking 1.579007e+12
## 2 0001368e-a25a-4541-92b0-057c79ff2ebd no_session_tracking 1.579465e+12
## 3 0001368e-a25a-4541-92b0-057c79ff2ebd no_session_tracking 1.579465e+12
## 4 0001368e-a25a-4541-92b0-057c79ff2ebd no_session_tracking 1.579465e+12
## 5 0001368e-a25a-4541-92b0-057c79ff2ebd no_session_tracking 1.579465e+12
## 6 0001368e-a25a-4541-92b0-057c79ff2ebd no_session_tracking 1.579465e+12
##              Level..Workspace.Id.                         Problem.Name
## 1 analyzing_models_2step_integers analyzing_models_2step_integers-069
## 2 analyzing_models_2step_integers analyzing_models_2step_integers-069
## 3 analyzing_models_2step_integers analyzing_models_2step_integers-069
## 4 analyzing_models_2step_integers analyzing_models_2step_integers-069
## 5 analyzing_models_2step_integers analyzing_models_2step_integers-069
## 6 analyzing_models_2step_integers analyzing_models_2step_integers-069
##            Step.Name Selection  Action  Input Outcome Help.Level
## 1 gn-classify-item-2           Attempt role-4     BUG          0
## 2 gn-classify-item-1           Attempt role-1     BUG          0
## 3 gn-classify-item-3           Attempt role-1      OK          0
## 4 gn-classify-item-3           Attempt role-1      OK          0
## 5 gn-classify-item-1           Attempt role-5     BUG          0
## 6 gn-classify-item-4           Attempt role-5      OK          0
##   Attempt.At.Step                                KC.Model.MATHia.
## 1               1         match _dep expression with description.
## 2               1 match _linear-term expression with description.
## 3               1   match _intercept expression with description.
## 4               1   match _intercept expression with description.
## 5               2 match _linear-term expression with description.
## 6               1                 interpret scenario with numbers
##      CF..Ruleid. CF..Etalon. CF..Skill.Previous.p.Known. CF..Skill.New.p.Known.
## 1 Match anything      role-3                   0.2010000              0.2458975
## 2 Match anything      role-4                   0.2010000              0.2458975
## 3                     role-1                   0.2010000              0.5587440
## 4                     role-1                   0.9020000              0.9852445
## 5 Match anything      role-4                          NA                     NA
## 6                     role-5                   0.9852445              0.9979243
##   CF..Workspace.Progress.Status.               CF..Semantic.Event.Id.
## 1                      GRADUATED dd263274-735e-45f7-a04c-337315b92744
## 2                      GRADUATED a85aebec-b8da-4525-a68b-7cb05d47bc68
## 3                      GRADUATED 330e96b9-ba06-4fde-a460-9aa10d3eaecb
## 4                      GRADUATED 330e96b9-ba06-4fde-a460-9aa10d3eaecb
## 5                      GRADUATED 998afd15-01d0-417d-9bc8-dc3cc33e1b77
## 6                      GRADUATED 666c76b8-7ae3-44b2-a3bf-9f196bf1bf83
```

```
head(scale_factor)
```

```
##                         Anon.Student.Id          Session.Id         Time
## 1 00003df3-4517-470e-a9e1-6d4a3a6d2f4e no_session_tracking 1.591637e+12
## 2 00003df3-4517-470e-a9e1-6d4a3a6d2f4e no_session_tracking 1.591637e+12
## 3 00003df3-4517-470e-a9e1-6d4a3a6d2f4e no_session_tracking 1.591637e+12
## 4 00003df3-4517-470e-a9e1-6d4a3a6d2f4e no_session_tracking 1.591637e+12
## 5 00003df3-4517-470e-a9e1-6d4a3a6d2f4e no_session_tracking 1.591637e+12
## 6 00003df3-4517-470e-a9e1-6d4a3a6d2f4e no_session_tracking 1.591637e+12
##   Level..Workspace.Id.             Problem.Name  Step.Name Selection
## 1     scale_drawings_3 scale_drawings_3_map-034 SF-value-n
## 2     scale_drawings_3 scale_drawings_3_map-034 SF-units-n
## 3     scale_drawings_3 scale_drawings_3_map-034 SF-value-d
## 4     scale_drawings_3 scale_drawings_3_map-034 SF-units-d
## 5     scale_drawings_3 scale_drawings_3_map-034 opt1-check
## 6     scale_drawings_3 scale_drawings_3_map-034 opt1-check
##              Action                                                Input
## 1           Attempt  {"escape-in-messages" : "false", "value" : "1"}
## 2           Attempt                                                cm
## 3           Attempt {"escape-in-messages" : "false", "value" : "40"}
## 4           Attempt                                                 m
## 5      Hint Request
## 6 Hint Level Change
##             Outcome Help.Level Attempt.At.Step
## 1                OK          0               1
## 2                OK          0               1
## 3                OK          0               1
## 4                OK          0               1
## 5      INITIAL_HINT          1               0
## 6 HINT_LEVEL_CHANGE          2               0
##                           KC.Model.MATHia.          CF..Ruleid. CF..Etalon.
## 1 scale-drawings-3-enter scale factor value.                                1
## 2 scale-drawings-3-enter scale factor units.                               cm
## 3 scale-drawings-3-enter scale factor value.                               40
## 4 scale-drawings-3-enter scale factor units.                                m
## 5                                           do the optional task        true
## 6
##   CF..Skill.Previous.p.Known. CF..Skill.New.p.Known.
## 1                   0.2010000             0.5587440
## 2                   0.6320000             0.9024567
## 3                   0.5587440             0.8444213
## 4                   0.9024567             0.9797706
## 5                          NA                    NA
## 6                          NA                    NA
##   CF..Workspace.Progress.Status.               CF..Semantic.Event.Id.
## 1                      GRADUATED 6676b24a-1fe7-42d9-a143-ef08a0b62afb
## 2                      GRADUATED 91126bd1-a762-4bfb-ab1f-df80fb45d124
## 3                      GRADUATED 23dd48aa-5884-43a6-bb6f-f76d5b0f0807
## 4                      GRADUATED 22a0852d-5069-4535-9c24-b7d50702d5f6
## 5                      GRADUATED 6932cd97-9d23-496b-b622-d82c1af1b0ff
## 6                      GRADUATED 30a57f3a-fba4-49e3-b86f-b59b373a34ab
```

Removing observations which have blank KCs.

```
new_scale_factor <- subset(scale_factor, KC.Model.MATHia. != "")
revised_new_scale_factor <- new_scale_factor %>%
  select(Anon.Student.Id, KC.Model.MATHia., Step.Name) %>%
  count(Anon.Student.Id, KC.Model.MATHia., Step.Name)
```

Transforming data by using `pivot_wider` to be able to compute correlations

```
opp_scale_factor <- revised_new_scale_factor %>%
  pivot_wider(names_from = KC.Model.MATHia., values_from = n)
```

We want each observation to have number of opportunities across KC.

```
new_opp_scale_factor <- opp_scale_factor %>%
  select(-Step.Name) %>%
  group_by(Anon.Student.Id) %>%
  mutate("scale-drawings-3-enter scale factor value." =
           list(`scale-drawings-3-enter scale factor value.`),
         "scale-drawings-3-enter scale factor units." =
           list(`scale-drawings-3-enter scale factor units.`),
         "scale-drawings-3-determine unknown measure, simple scale factor." =
           list(`scale-drawings-3-determine unknown measure, simple scale factor.`),
         "scale-drawings-3-determine unknown measure, complex scale factor." =
           list(`scale-drawings-3-determine unknown measure, complex scale factor.`)) %>%
  distinct()
```

```
head(new_opp_scale_factor)
```

```
## # A tibble: 6 x 5
## # Groups:   Anon.Student.Id [6]
##   Anon.Student.Id `scale-drawings~ `scale-drawings~ `scale-drawings~
##   <chr>           <list>           <list>           <list>
## 1 00003df3-4517-~ <int [6]>        <int [6]>        <int [6]>
## 2 0000bede-d927-~ <int [6]>        <int [6]>        <int [6]>
## 3 00027f1f-dd6d-~ <int [6]>        <int [6]>        <int [6]>
## 4 0003fb96-4a01-~ <int [6]>        <int [6]>        <int [6]>
## 5 00052b99-8847-~ <int [6]>        <int [6]>        <int [6]>
## 6 0005d5fd-cdce-~ <int [6]>        <int [6]>        <int [6]>
## # ... with 1 more variable: `scale-drawings-3-enter scale factor value.` <list>
```

Reating the above procedure for another workspace. We want each observation to have number of opportunities across KC.

```
new_model_2_factor <- subset(model_2_factor, KC.Model.MATHia. != "")
revised_new_model_2_factor <- new_model_2_factor %>%
  select(Anon.Student.Id, KC.Model.MATHia., Step.Name) %>%
  count(Anon.Student.Id, KC.Model.MATHia., Step.Name)
```

```
opp_model_2_factor <- revised_new_model_2_factor %>%
  pivot_wider(names_from = KC.Model.MATHia., values_from = n)
```

```
new_opp_model_2_factor <- opp_model_2_factor %>%
  select(-Step.Name) %>%
```

```
  group_by(Anon.Student.Id) %>%
  mutate("identifying units-1" = list(`identifying units-1`),
         "enter given, reading words-1" = list(`enter given, reading words-1`),
         "find y, any form-1" = list(`find y, any form-1`),
         "enter given, reading numerals-1" = list(`enter given, reading numerals-1`),
         "define variable-1" = list(`define variable-1`),
         "write expression, positive intercept-1" =
           list(`write expression, positive intercept-1`),
         "write expression, negative intercept-1" =
           list(`write expression, negative intercept-1`),
         "write expression, negative slope-1" =
           list(`write expression, negative slope-1`),
         "write expression, positive slope-1" =
           list(`write expression, positive slope-1`)) %>%
  distinct()
```

```
head(new_opp_model_2_factor)
```

```
## # A tibble: 6 x 10
## # Groups:   Anon.Student.Id [6]
##   Anon.Student.Id `define variabl~ `enter given, r~ `enter given, r~
##   <chr>           <list>           <list>           <list>
## 1 0001368e-a25a-~ <int [8]>        <int [8]>        <int [8]>
## 2 00027f1f-dd6d-~ <int [8]>        <int [8]>        <int [8]>
## 3 00052b99-8847-~ <int [8]>        <int [8]>        <int [8]>
## 4 0005d5fd-cdce-~ <int [8]>        <int [8]>        <int [8]>
## 5 000a2b86-ace0-~ <int [8]>        <int [8]>        <int [8]>
## 6 000b1183-eb07-~ <int [8]>        <int [8]>        <int [8]>
## # ... with 6 more variables: `find y, any form-1` <list>, `identifying
## #   units-1` <list>, `write expression, negative intercept-1` <list>, `write
## #   expression, negative slope-1` <list>, `write expression, positive
## #   intercept-1` <list>, `write expression, positive slope-1` <list>
```

Reating the above procedure for another workspace. We want each observation to have number of opportunities across KC.

```
new_analyze_2_factor <- subset(analyze_2_factor, KC.Model.MATHia. != "")
revised_new_analyze_2_factor <- new_analyze_2_factor %>%
  select(Anon.Student.Id, KC.Model.MATHia., Step.Name) %>%
  count(Anon.Student.Id, KC.Model.MATHia., Step.Name)
```

```
opp_analyze_2_factor <- revised_new_analyze_2_factor %>%
  pivot_wider(names_from = KC.Model.MATHia., values_from = n)
```

```
new_opp_analyze_2_factor <- opp_analyze_2_factor %>%
  select(-Step.Name) %>%
  group_by(Anon.Student.Id) %>%
  mutate("match _dep expression with description." =
           list(`match _dep expression with description.`),
         "match _linear-term expression with description." =
           list(`match _linear-term expression with description.`),
         "match _intercept expression with description." =
           list(`match _intercept expression with description.`),
         "interpret scenario with numbers" =
```

```
          list(`interpret scenario with numbers`),
        "match _indep expression with description." =
          list(`match _indep expression with description.`),
        "interpret scenario with words" =
          list(`interpret scenario with words`),
        "match _slope expression with description." =
          list(`match _slope expression with description.`)) %>%
  distinct()
```

```
head(new_opp_analyze_2_factor)
```

```
## # A tibble: 6 x 8
## # Groups:   Anon.Student.Id [6]
##   Anon.Student.Id `interpret scen~ `interpret scen~ `match _dep exp~
##   <chr>           <list>           <list>           <list>
## 1 0001368e-a25a-~ <int [6]>        <int [6]>        <int [6]>
## 2 00026214-e84d-~ <int [6]>        <int [6]>        <int [6]>
## 3 00027f1f-dd6d-~ <int [6]>        <int [6]>        <int [6]>
## 4 0003fb96-4a01-~ <int [6]>        <int [6]>        <int [6]>
## 5 00052b99-8847-~ <int [6]>        <int [6]>        <int [6]>
## 6 0005d5fd-cdce-~ <int [6]>        <int [6]>        <int [6]>
## # ... with 4 more variables: `match _indep expression with
## #   description.` <list>, `match _intercept expression with
## #   description.` <list>, `match _linear-term expression with
## #   description.` <list>, `match _slope expression with description.` <list>
```

Joining these dataframes by `Anon.Student.Id`

```
opp_all_workspaces <- new_opp_analyze_2_factor %>%
  dplyr::inner_join(new_opp_model_2_factor, by = "Anon.Student.Id")
opp_all_workspaces <- opp_all_workspaces %>%
  dplyr::inner_join(new_opp_scale_factor, by = "Anon.Student.Id")
```

Remove NAs for each cell.

```
new_opp_all_workspaces <- opp_all_workspaces
remove_na <- function(v){
  v1 <- v[!is.na(v)]
  if (is_empty(v1) == TRUE){
    return(0)
  }
  return(v1)
}
for (i in colnames(new_opp_all_workspaces)[-1]){
  new_opp_all_workspaces[[i]] <- lapply(new_opp_all_workspaces[[i]], remove_na)
}
new_opp_all_workspaces <-
  new_opp_all_workspaces[,order(colnames(new_opp_all_workspaces))]
```

```
# Sanity check
check_1 <- new_opp_all_workspaces
for (i in colnames(check_1)[-1]){
  check_1[[i]] <- lapply(check_1[[i]], sum)
}
check_cols <- colnames(check_1)[1]
```

```
for (i in colnames(check_1)[-1]){
  check_cols <- c(check_cols, substr(i,7,100))
}
colnames(check_1) <- check_cols
check_1 <- check_1[,order(colnames(check_1))]
```

```
# Sanity check
check_2_scale_factor <- new_scale_factor %>%
  select(Anon.Student.Id, KC.Model.MATHia., Outcome) %>%
  group_by(Anon.Student.Id)
check_2_scale_factor <- check_2_scale_factor %>%
  pivot_wider(names_from = KC.Model.MATHia., values_from = Outcome)
check_2_model_2_factor <- new_model_2_factor %>%
  select(Anon.Student.Id, KC.Model.MATHia., Outcome) %>%
  group_by(Anon.Student.Id)
check_2_model_2_factor <- check_2_model_2_factor %>%
  pivot_wider(names_from = KC.Model.MATHia., values_from = Outcome)
check_2_analyze_2_factor <- new_analyze_2_factor %>%
  select(Anon.Student.Id, KC.Model.MATHia., Outcome) %>%
  group_by(Anon.Student.Id)
check_2_analyze_2_factor <- check_2_analyze_2_factor %>%
  pivot_wider(names_from = KC.Model.MATHia., values_from = Outcome)
check_2 <- check_2_analyze_2_factor %>%
  dplyr::inner_join(check_2_model_2_factor, by = "Anon.Student.Id")
check_2 <- check_2 %>%
  dplyr::inner_join(check_2_scale_factor, by = "Anon.Student.Id")
check_2 <- check_2[,order(colnames(check_2))]
opp_data <- check_2
for (i in colnames(check_2)[-1]){
  check_2[[i]] <- lapply(check_2[[i]], length)
}
```

Outcome data and opportunity count per KC

```
full_opp <- cbind(new_opp_all_workspaces, opp_data)
```

Function for obtaining success rate for each opportunities.

```
get_opp_success_rate <- function(v1, v2){
  if (v1 == 0 | is.null(v2)==TRUE){
    return(0)
  }
  success <- function(vec){
    ret_vec <- length(which(vec %in% "OK"))/length(vec)
    return(ret_vec)
  }
  ret_fin_vec <- c()
  v <- v2
  for (i in 1:length(v1)){
    ret_fin_vec <- c(ret_fin_vec, success(v[1:v1[i]]))
    v <- v[-c(1:v1[i])]
  }
  return(ret_fin_vec)
}
```

Testing the function.

```
get_opp_success_rate(0, c("OK","ERROR","OK","OK", "BUG"))
```

```
## [1] 0
```

```
get_opp_success_rate(c(3,2), c("OK","ERROR","OK","OK", "BUG"))
```

```
## [1] 0.6666667 0.5000000
```

```
get_opp_success_rate(c(2,3), c("OK","ERROR","OK","OK", "BUG"))
```

```
## [1] 0.5000000 0.6666667
```

```
get_opp_success_rate(c(1,1,3), c("OK","ERROR","OK","OK", "BUG"))
```

```
## [1] 1.0000000 0.0000000 0.6666667
```

Creating list having success rate across opportunities for KC for each student.

```
opp_opp <- list()
for (j in 2:21){
  temp <- full_opp[,c(j,j+21)]
  x <- c(temp[[1]], temp[[2]])
  temp_res <- list()
  for (i in 1:13114){
    temp_res[[i]] <- get_opp_success_rate(x[[i]],x[[i+13114]])
  }
  opp_opp[[j-1]] <- temp_res
}
```

Creating a dataframe.

```
tp_opp <- new_opp_all_workspaces
for (i in 1:20){
  tp_opp[[i+1]] <- as.vector(opp_opp[[i]])
}
```

Appendix 2

In this technical appendix, we import the dataframe created in technical appendix 1 and build a function to build a Gaussian Graphical Model. The inputs to the function are `data_frame` (in the format shown in examples), `initial_opportunities` (example: 1,2,3,..), and `minimum_correlateion` (GGM would show lines having absolute value of partial correlation above this threshold).

Loading packages.

```
require(tidyverse)
require(qgraph)
require(xtable)
require(dplyr)
require(bootnet)
require(rstudioapi)
```

Creating logit function.

```
logit <- function(v){
  if (is.na(v)==TRUE){
    return(NA)
  }
  if (is.nan(v) == TRUE){
    return(NA)
  }
  if (v==1){
    v = 0.9999
  }
  if (v==0){
    v = 0.0001
  }
  return(log(v/(1-v)))
}
```

Removing `Anon.Student.Id` from the dataset.

```
ggm_opp <- tp_opp[-1]
```

Function for creating GGM for KCs.

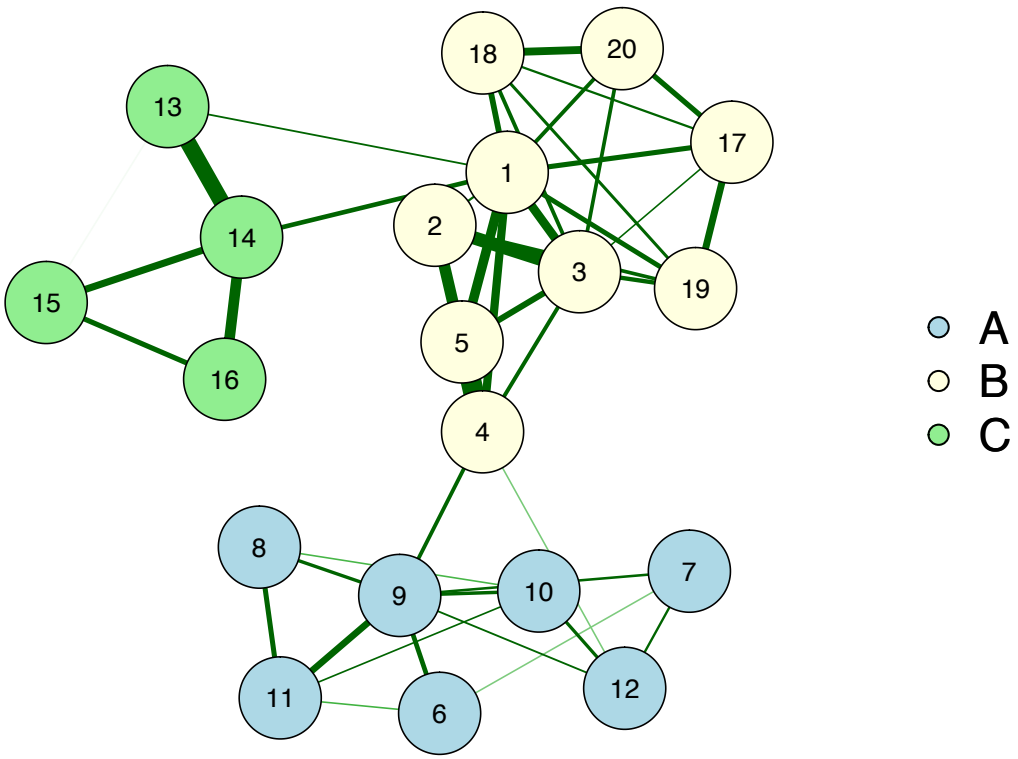
```
ggm <- function(dat, num, min_cor = 0.03){
  success_initial_opp <- function(v){
      if(length(v) >= num){
        v1 <- as.numeric(v[1:num])
        success <- sum(v1)/num
        return(success)
```

```
      }
      else {v1 <- as.numeric(v)}
      success <- sum(v1)/length(v1)
      return(success)
    }
  opp_data_success_initial <- dat
  for (i in colnames(opp_data_success_initial)){
    opp_data_success_initial[[i]] <-
      lapply(opp_data_success_initial[[i]], success_initial_opp)
  }
  new_opp_data_success_initial <- opp_data_success_initial
  for (i in colnames(new_opp_data_success_initial)){
      new_opp_data_success_initial[[i]] <-
        lapply(new_opp_data_success_initial[[i]], logit)
  }
  for (i in 1:ncol(new_opp_data_success_initial)){
      new_opp_data_success_initial[[i]] <-
        as.numeric(new_opp_data_success_initial[[i]])
  }
  new_opp_data_success_initial <- na.omit(new_opp_data_success_initial)
  data_success_initial_corr <- psych::corFiml(new_opp_data_success_initial)
  group_items <- list(
    A = c(6:12),
    B = c(1:5,17:20),
    C = c(13:16)
    )
  return(qgraph::qgraph(
      data_success_initial_corr,
      layout = "spring",
      graph = "glasso",
      labels = TRUE,
      legend.cex = 0.7,
      tuning = 0.1,
      color = c("light blue", "light yellow", "light green"),
      groups = group_items,
      labels = TRUE,
      sampleSize = nrow(new_opp_data_success_initial),
      minimum = min_cor
    ))
}
```

In the following plots, for the sake of easier visual comparison, we only show partial correlations that are larger than 0.05. In this way, only lines representing stronger partial correlations would be shown.

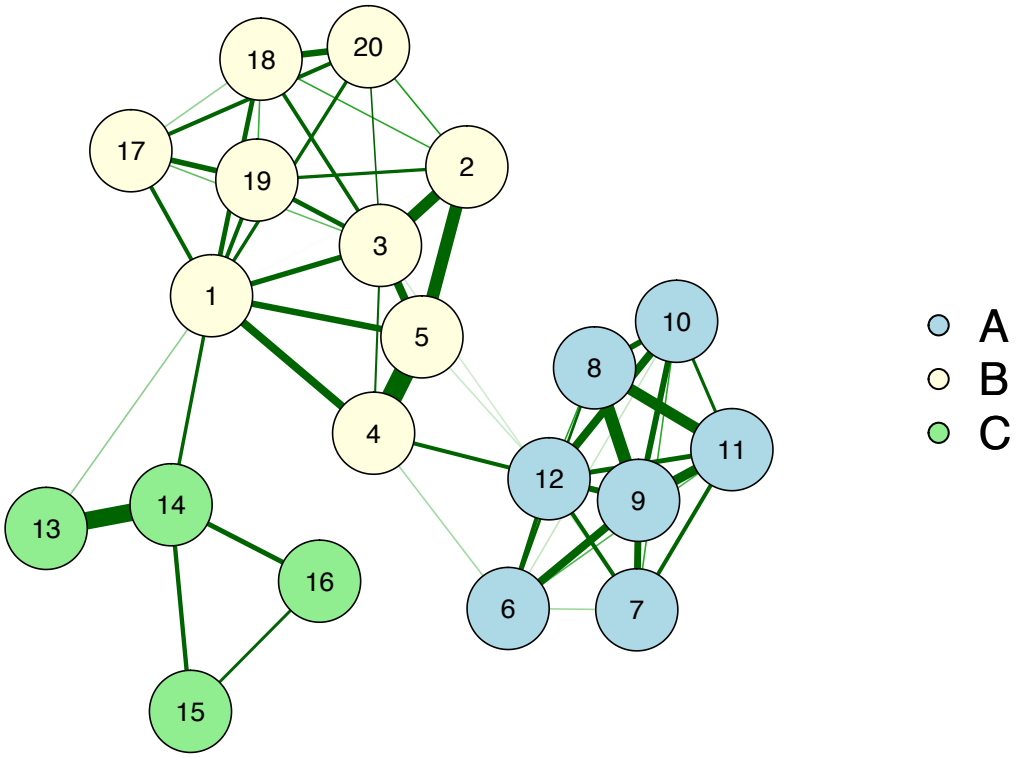
When the number of initial opportunities is 1, the GGM plot is as following:

```
x2 <- ggm(dat = ggm_opp,num = 1,min_cor = 0.05)
```



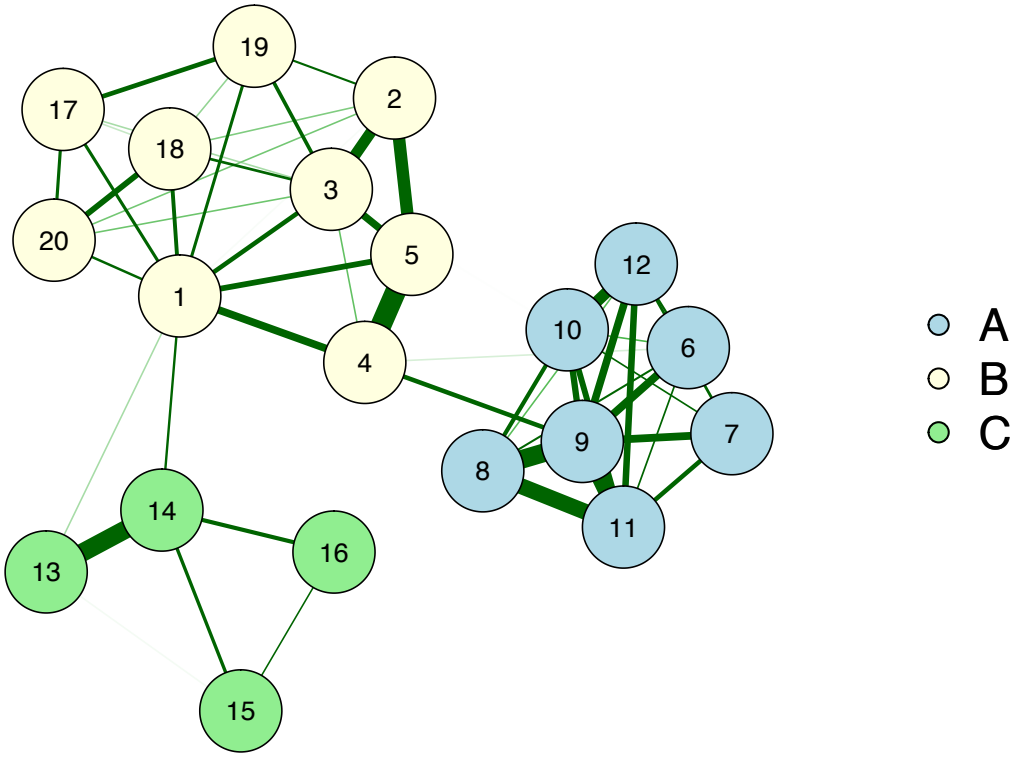
When the number of initial opportunities is 2, the GGM plot is as following:

```
ggm(dat = ggm_opp,num = 2,min_cor = 0.05)
```



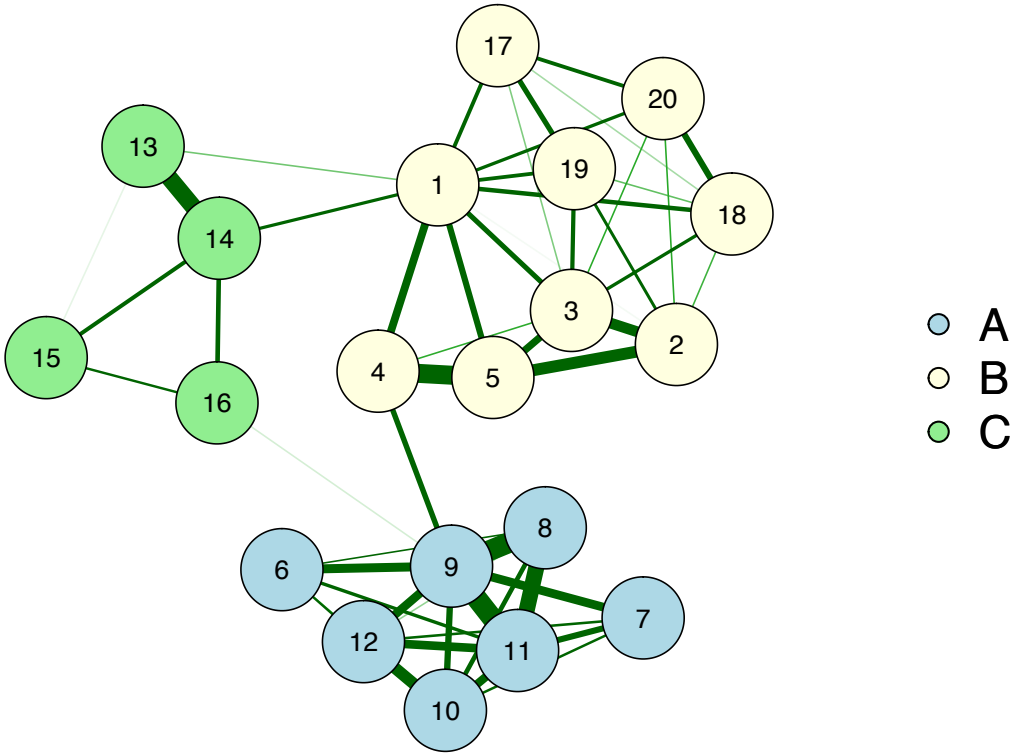
When the number of initial opportunities is 3, the GGM plot is as following:

```
ggm(dat = ggm_opp,num = 3,min_cor = 0.05)
```



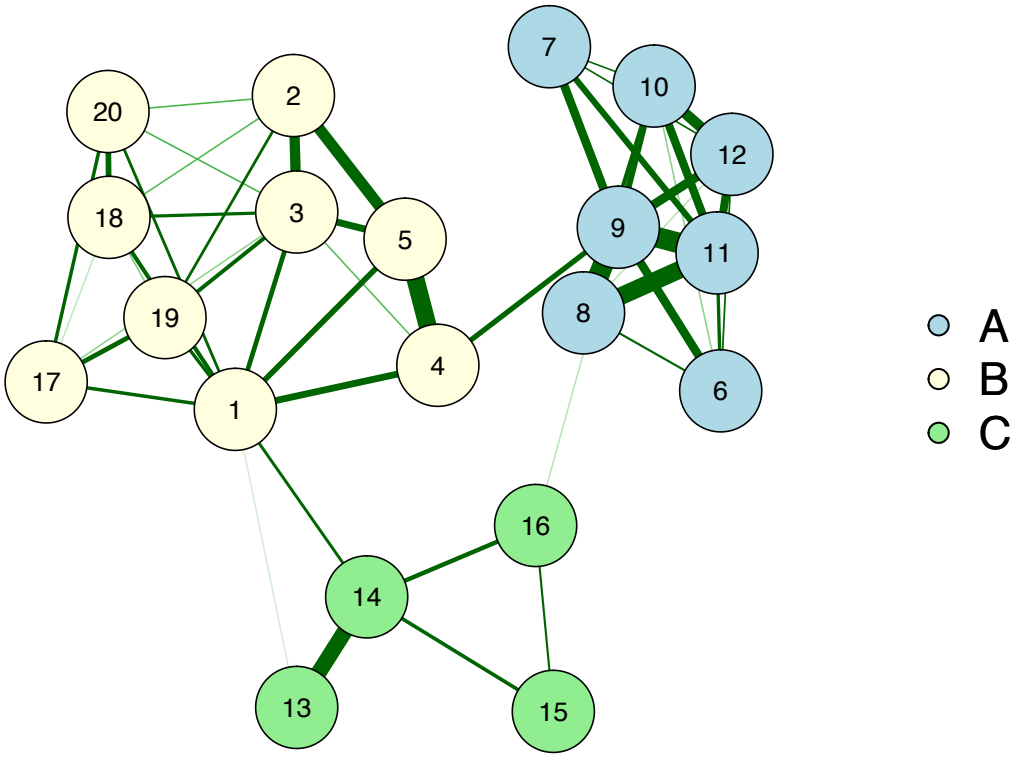
When the number of initial opportunities is 4, the GGM plot is as following:

```
ggm(dat = ggm_opp,num = 4,min_cor = 0.05)
```



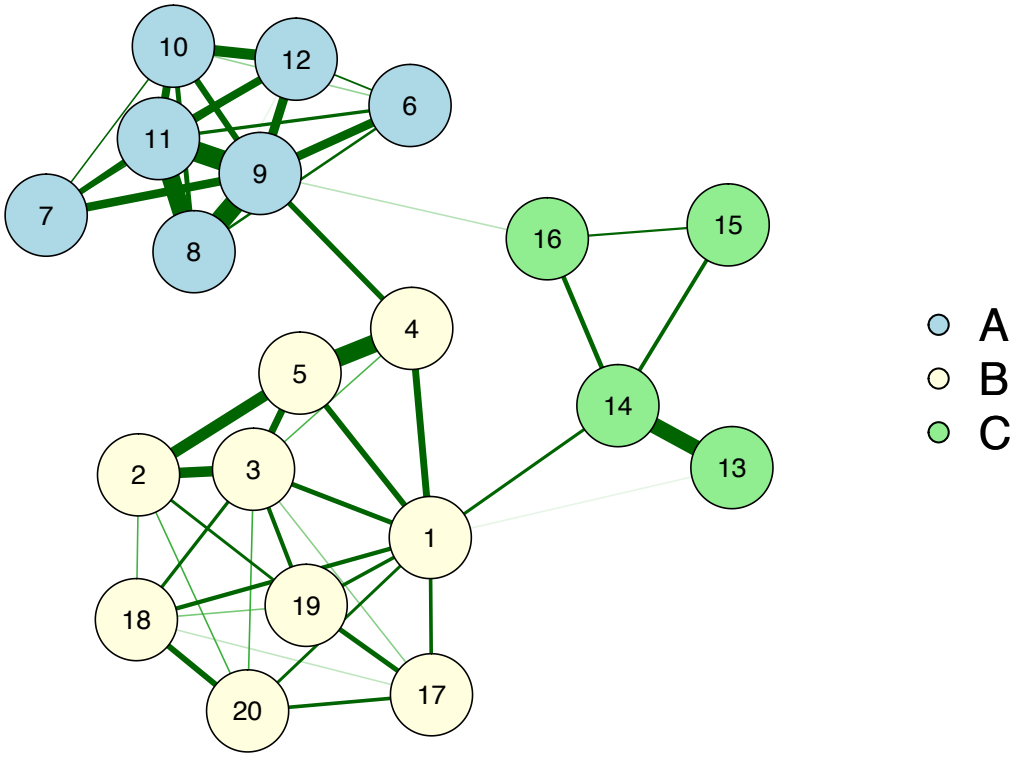
When the number of initial opportunities is 5, the GGM plot is as following:

```
ggm(dat = ggm_opp,num = 5,min_cor = 0.05)
```



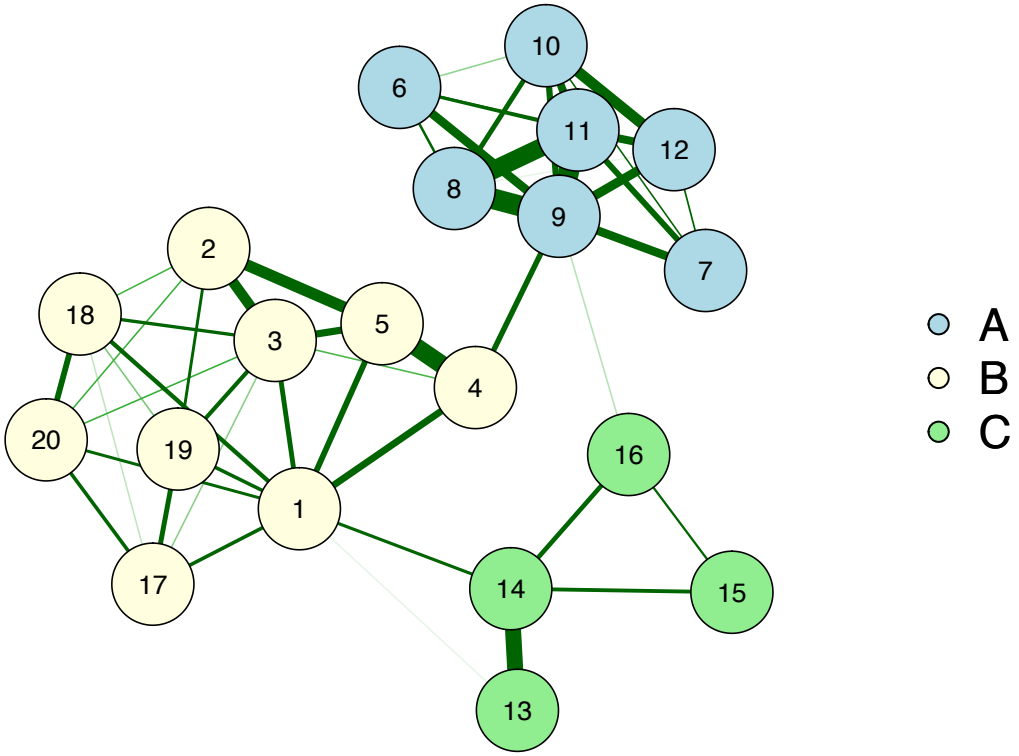
When the number of initial opportunities is 6, the GGM plot is as following:

```
ggm(dat = ggm_opp,num = 6,min_cor = 0.05)
```



When the number of initial opportunities is 7, the GGM plot is as following:

```
ggm(dat = ggm_opp,num = 7,min_cor = 0.05)
```



As we can see, when the number of initial opportunities is 2, the partial correlations between workspace A and workspace B are the strongest. There are more lines between those two workspaces. Therefore, we chose 2 as our final cutoff point for initial opportunities.

```
df2 <- data.frame(x2$Edgelist$from, x2$Edgelist$to, x2$Edgelist$weight)
df2 <- df2 %>%
  arrange(desc(x2$Edgelist$weight))
head(df2, n = 10)
```

```
##    x2.Edgelist.from x2.Edgelist.to x2.Edgelist.weight
## 1                 4               5         0.19003561
## 2                13              14         0.15727573
## 3                 2               3         0.13730900
## 4                 2               5         0.12467583
## 5                14              16         0.11857221
## 6                 1               5         0.11109871
## 7                 1               3         0.10539173
## 8                 1               4         0.10149248
## 9                18              20         0.09862264
## 10               14              15         0.09364421
```

Appendix 3

This appendix contains the code for GLMER method (prerequisite relations between KCs) and retrospecitve study two-sample T test (prerequisite relations between workspaces).

Generalized Linear Mixed-Effects Model (glmer)

Initial Data Processing

```
library(tidyverse)
library(ggplot2)
library(lme4)
```

Read in and process workspace transaction data.

```
b = read.delim("b.txt", header = T)
#Only keep records where KC is not null
b$KC.Model.MATHia. = ifelse(b$KC.Model.MATHia. == "", NA, b$KC.Model.MATHia.)
b = b[!is.na(b$KC.Model.MATHia.),]
#Get rid of unneccessary columns
b_clean = b[, c(1, 3, 5, 6, 8, 10, 11, 12, 13, 17)]
```

Read in and process workspace step rollup data

```
b_student = read.delim("b_student.txt", header = T)
#keep records where KC is not null
b_student$KC..MATHia. = ifelse(b_student$KC..MATHia. == "",
                               NA, b_student$KC..MATHia.)
b_student = b_student[!is.na(b_student$KC..MATHia.),]
#Get rid of unneccessary columns and recode first attempt column
b_student_clean = b_student[, c(3, 4, 5, 7, 8, 15, 20, 21)]
b_student_clean$First.Attempt = ifelse(b_student_clean$First.Attempt ==
                                       "correct", 1, 0)
```

Modeling

There are 9 unique KCs in workspace B

```
unique(b_student_clean$KC..MATHia.)
```

```
## [1] "identifying units-1"
## [2] "enter given, reading numerals-1"
## [3] "define variable-1"
## [4] "find y, any form-1"
## [5] "write expression, positive intercept-1"
## [6] "enter given, reading words-1"
## [7] "write expression, negative slope-1"
## [8] "write expression, positive slope-1"
## [9] "write expression, negative intercept-1"
```

Merge two datasets and add indicators of whether student has mastered KC1 and KC2

```
kc.1 = "find y, any form-1"
kc.2 = "identifying units-1"

df_all = NA

for (id in unique(b_student_clean$Anon.Student.Id)){
  temp.1 = b_clean[which(b_clean$Anon.Student.Id == id &
                         b_clean$KC.Model.MATHia.%in% c(kc.1, kc.2) &
                           b_clean$Attempt.At.Step == 1), ]
  temp.1 = unique(temp.1)
  temp.1 = temp.1[order(temp.1$Time), ]
  temp.1 = unique(temp.1[, -2])

  temp.2 = b_student_clean[which(b_student_clean$Anon.Student.Id == id &
                               b_student_clean$KC..MATHia. %in% c(kc.1, kc.2)), ]

  if (nrow(temp.1) != nrow(temp.2)){
    next
  }

  temp.2$CF = temp.1$CF..Skill.New.p.Known.

  df = temp.2[, c(1, 6, 7, 8, 9)]
  df$CF.ind = ifelse(df$CF > 0.95, 1, 0)

  if (length(which(df$KC..MATHia. == kc.2 & df$CF.ind == 1)) == 0){
    next
  }
  if (length(which(df$KC..MATHia. == kc.1 & df$CF.ind == 1)) == 0){
    next
  }

  if (which(df$KC..MATHia. == kc.2 & df$CF.ind == 1)[1]-1 == 0){
    df$know_kc2 = rep(1, nrow(df)+1-which(df$KC..MATHia. == kc.2 & df$CF.ind == 1)[1])
  }
  else{
    df$know_kc2 = c(rep(0, which(df$KC..MATHia. == kc.2 & df$CF.ind == 1)[1]-1),
                    rep(1, nrow(df)+1-which(df$KC..MATHia. == kc.2 & df$CF.ind == 1)[1]))
  }

  if (which(df$KC..MATHia. == kc.1 & df$CF.ind == 1)[1]-1 == 0){
    df$know_kc1 = rep(1, nrow(df)+1-which(df$KC..MATHia. == kc.1 & df$CF.ind == 1)[1])
  }
  else{
    df$know_kc1 = c(rep(0, which(df$KC..MATHia. == kc.1 & df$CF.ind == 1)[1]-1),
                    rep(1, nrow(df)+1-which(df$KC..MATHia. == kc.1 & df$CF.ind == 1)[1]))
  }

  if (is.na(df_all)){
    df_all = df
  }
  else{
    df_all = rbind(df_all, df)
```

```
  }

}
```

Run Glmer on KC1 and KC2

```
df.kc1 = df_all[which(df_all$KC..MATHia. == kc.1), ]
fit.1 = glmer(First.Attempt ~ 1 + Opportunity..MATHia. +
              Opportunity..MATHia. : know_kc2 + know_kc2 + (1|Anon.Student.Id),
            data = df.kc1, family = "binomial")


df.kc2 = df_all[which(df_all$KC..MATHia. == kc.2), ]
fit.2 = glmer(First.Attempt ~ 1 + Opportunity..MATHia. +
              Opportunity..MATHia. : know_kc1 + know_kc1 + (1|Anon.Student.Id),
            data = df.kc2, family = "binomial")

summary(fit.1)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula:
## First.Attempt ~ 1 + Opportunity..MATHia. + Opportunity..MATHia.:know_kc2 +
##     know_kc2 + (1 | Anon.Student.Id)
##    Data: df.kc1
##
##      AIC      BIC   logLik deviance df.resid
##  15682.6  15719.8  -7836.3  15672.6    12569
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.6520 -0.8762  0.4062  0.7836  3.2220
##
## Random effects:
##  Groups          Name        Variance Std.Dev.
##  Anon.Student.Id (Intercept) 0.83     0.9111
## Number of obs: 12574, groups:  Anon.Student.Id, 465
##
## Fixed effects:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                   -0.582045   0.069934  -8.323  < 2e-16 ***
## Opportunity..MATHia.           0.076786   0.007044  10.901  < 2e-16 ***
## know_kc2                       0.496252   0.077558   6.398 1.57e-10 ***
## Opportunity..MATHia.:know_kc2 -0.040398   0.007040  -5.738 9.56e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) Op..MATH. knw_k2
## Oppr..MATH. -0.579
## know_kc2    -0.545  0.400
## O..MATH.:_2  0.562 -0.926    -0.633
## optimizer (Nelder_Mead) convergence code: 0 (OK)
```

```
## Model is nearly unidentifiable: very large eigenvalue
##  - Rescale variables?
```

```
summary(fit.2)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula:
## First.Attempt ~ 1 + Opportunity..MATHia. + Opportunity..MATHia.:know_kc1 +
##     know_kc1 + (1 | Anon.Student.Id)
##    Data: df.kc2
##
##      AIC      BIC   logLik deviance df.resid
##   8898.1   8933.0  -4444.1   8888.1     7940
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.1285 -0.7912  0.4208  0.6060  2.4517
##
## Random effects:
##  Groups          Name        Variance Std.Dev.
##  Anon.Student.Id (Intercept) 0.6916   0.8316
## Number of obs: 7945, groups:  Anon.Student.Id, 465
##
## Fixed effects:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  -0.17276    0.07573  -2.281   0.0225 *
## Opportunity..MATHia.          0.10421    0.01179   8.836  < 2e-16 ***
## know_kc1                      1.20321    0.10633  11.316  < 2e-16 ***
## Opportunity..MATHia.:know_kc1 -0.06630    0.01208  -5.490 4.02e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) Op..MATH. knw_k1
## Oppr..MATH. -0.642
## know_kc1    -0.493  0.337
## O..MATH.:_1  0.598 -0.868    -0.674
```

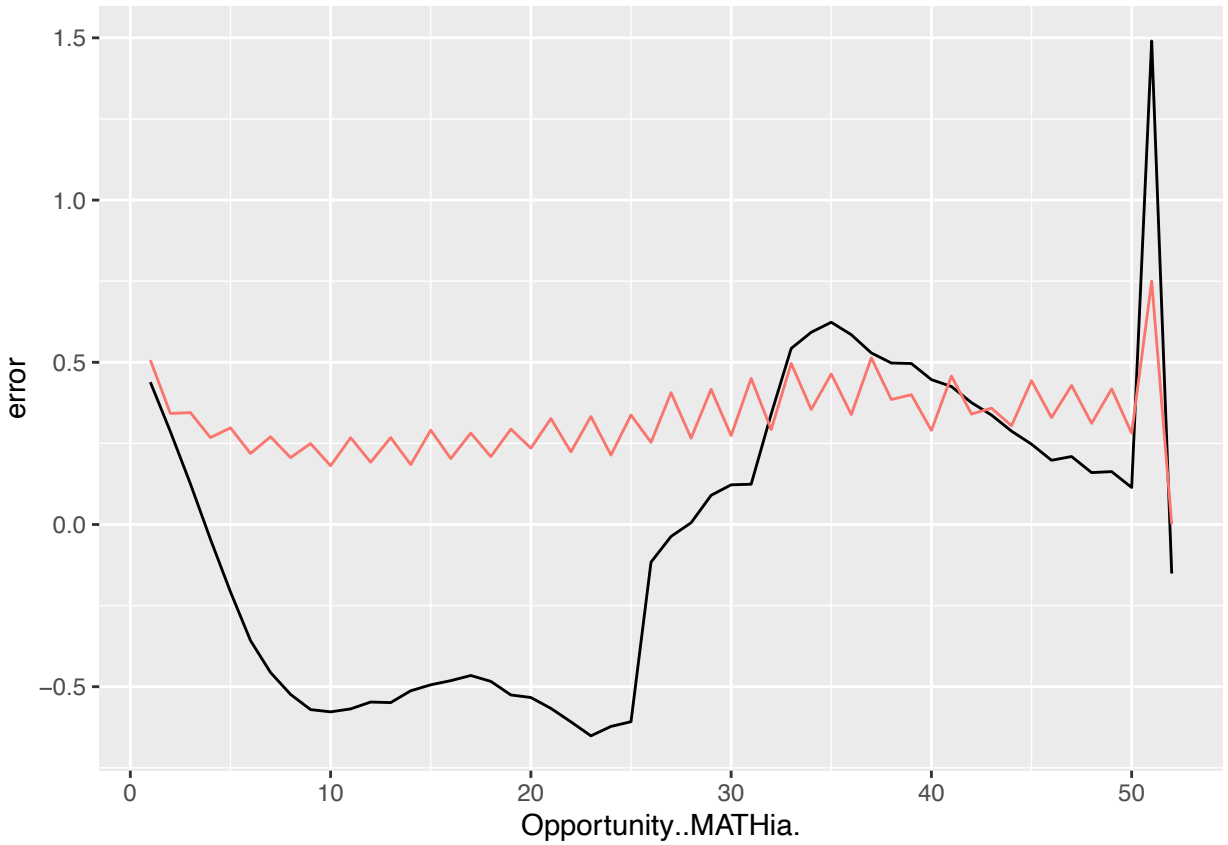
Overall Model (Not used)

Overall glm/glmer model of learning curve

```
fit.glmer <- glmer(First.Attempt ~ KC..MATHia. + KC..MATHia.:Opportunity..MATHia. +
                (1|Anon.Student.Id), data = b_student_clean, family = "binomial")
fit.glm <- glm(First.Attempt ~ KC..MATHia. + KC..MATHia.:Opportunity..MATHia.,
             data = b_student_clean)
```

Model with mixed effects (student ID)

```
b_student_clean$predicted = predict(fit.glmer)
error_rate = b_student_clean %>%
  group_by(Opportunity..MATHia.) %>%
  summarise(error = 1 - mean(predicted), actual = 1 - sum(First.Attempt)/n())
ggplot(error_rate, aes(x = Opportunity..MATHia.)) + geom_line(aes(y = error)) +
  geom_line(aes(y = actual, color = "red")) + theme(legend.position = "none")
```




```
summary(fit.glmer)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: First.Attempt ~ KC..MATHia. + KC..MATHia.:Opportunity..MATHia. +
##     (1 | Anon.Student.Id)
##    Data: b_student_clean
##
##      AIC      BIC   logLik deviance df.resid
##  47298.9  47466.3 -23630.5  47260.9    49437
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -16.0024  -0.6264   0.2810   0.5817  21.8550
##
## Random effects:
##  Groups          Name        Variance Std.Dev.
##  Anon.Student.Id (Intercept) 0.983    0.9915
## Number of obs: 49456, groups:  Anon.Student.Id, 500
##
## Fixed effects:
##                                                                  Estimate
## (Intercept)                                                      1.489464
## KC..MATHia.enter given, reading numerals-1                       0.632543
## KC..MATHia.enter given, reading words-1                          0.174637
## KC..MATHia.find y, any form-1                                   -1.900331
```

```
## KC..MATHia.identifying units-1                                          -1.394316
## KC..MATHia.write expression, negative intercept-1                       -1.924184
## KC..MATHia.write expression, negative slope-1                           -1.990265
## KC..MATHia.write expression, positive intercept-1                       -1.748999
## KC..MATHia.write expression, positive slope-1                           -1.395225
## KC..MATHia.define variable-1:Opportunity..MATHia.                        0.159022
## KC..MATHia.enter given, reading numerals-1:Opportunity..MATHia.          0.099882
## KC..MATHia.enter given, reading words-1:Opportunity..MATHia.             0.086912
## KC..MATHia.find y, any form-1:Opportunity..MATHia.                       0.049593
## KC..MATHia.identifying units-1:Opportunity..MATHia.                      0.092602
## KC..MATHia.write expression, negative intercept-1:Opportunity..MATHia.   0.295517
## KC..MATHia.write expression, negative slope-1:Opportunity..MATHia.       0.243034
## KC..MATHia.write expression, positive intercept-1:Opportunity..MATHia.   0.152690
## KC..MATHia.write expression, positive slope-1:Opportunity..MATHia.       0.099242
##                                                                        Std. Error
## (Intercept)                                                              0.084208
## KC..MATHia.enter given, reading numerals-1                               0.104275
## KC..MATHia.enter given, reading words-1                                  0.097230
## KC..MATHia.find y, any form-1                                            0.078540
## KC..MATHia.identifying units-1                                           0.083332
## KC..MATHia.write expression, negative intercept-1                        0.133533
## KC..MATHia.write expression, negative slope-1                            0.120381
## KC..MATHia.write expression, positive intercept-1                        0.110108
## KC..MATHia.write expression, positive slope-1                            0.120479
## KC..MATHia.define variable-1:Opportunity..MATHia.                        0.008940
## KC..MATHia.enter given, reading numerals-1:Opportunity..MATHia.          0.007991
## KC..MATHia.enter given, reading words-1:Opportunity..MATHia.             0.007060
## KC..MATHia.find y, any form-1:Opportunity..MATHia.                       0.001709
## KC..MATHia.identifying units-1:Opportunity..MATHia.                      0.003653
## KC..MATHia.write expression, negative intercept-1:Opportunity..MATHia.   0.040980
## KC..MATHia.write expression, negative slope-1:Opportunity..MATHia.       0.028478
## KC..MATHia.write expression, positive intercept-1:Opportunity..MATHia.   0.020431
## KC..MATHia.write expression, positive slope-1:Opportunity..MATHia.       0.028400
##                                                                          z value
## (Intercept)                                                               17.688
## KC..MATHia.enter given, reading numerals-1                                 6.066
## KC..MATHia.enter given, reading words-1                                    1.796
## KC..MATHia.find y, any form-1                                            -24.196
## KC..MATHia.identifying units-1                                           -16.732
## KC..MATHia.write expression, negative intercept-1                        -14.410
## KC..MATHia.write expression, negative slope-1                            -16.533
## KC..MATHia.write expression, positive intercept-1                        -15.884
## KC..MATHia.write expression, positive slope-1                            -11.581
## KC..MATHia.define variable-1:Opportunity..MATHia.                         17.787
## KC..MATHia.enter given, reading numerals-1:Opportunity..MATHia.           12.500
## KC..MATHia.enter given, reading words-1:Opportunity..MATHia.              12.311
## KC..MATHia.find y, any form-1:Opportunity..MATHia.                        29.017
## KC..MATHia.identifying units-1:Opportunity..MATHia.                       25.352
## KC..MATHia.write expression, negative intercept-1:Opportunity..MATHia.    7.211
## KC..MATHia.write expression, negative slope-1:Opportunity..MATHia.        8.534
## KC..MATHia.write expression, positive intercept-1:Opportunity..MATHia.    7.473
## KC..MATHia.write expression, positive slope-1:Opportunity..MATHia.        3.494
##                                                                         Pr(>|z|)
## (Intercept)                                                              < 2e-16
```

```
## KC..MATHia.enter given, reading numerals-1                                  1.31e-09
## KC..MATHia.enter given, reading words-1                                     0.072475
## KC..MATHia.find y, any form-1                                                < 2e-16
## KC..MATHia.identifying units-1                                               < 2e-16
## KC..MATHia.write expression, negative intercept-1                            < 2e-16
## KC..MATHia.write expression, negative slope-1                                < 2e-16
## KC..MATHia.write expression, positive intercept-1                            < 2e-16
## KC..MATHia.write expression, positive slope-1                                < 2e-16
## KC..MATHia.define variable-1:Opportunity..MATHia.                            < 2e-16
## KC..MATHia.enter given, reading numerals-1:Opportunity..MATHia.              < 2e-16
## KC..MATHia.enter given, reading words-1:Opportunity..MATHia.                 < 2e-16
## KC..MATHia.find y, any form-1:Opportunity..MATHia.                           < 2e-16
## KC..MATHia.identifying units-1:Opportunity..MATHia.                          < 2e-16
## KC..MATHia.write expression, negative intercept-1:Opportunity..MATHia. 5.54e-13
## KC..MATHia.write expression, negative slope-1:Opportunity..MATHia.           < 2e-16
## KC..MATHia.write expression, positive intercept-1:Opportunity..MATHia. 7.82e-14
## KC..MATHia.write expression, positive slope-1:Opportunity..MATHia.      0.000475
##
## (Intercept)                                                                 ***
## KC..MATHia.enter given, reading numerals-1                                  ***
## KC..MATHia.enter given, reading words-1                                     .
## KC..MATHia.find y, any form-1                                               ***
## KC..MATHia.identifying units-1                                              ***
## KC..MATHia.write expression, negative intercept-1                           ***
## KC..MATHia.write expression, negative slope-1                               ***
## KC..MATHia.write expression, positive intercept-1                           ***
## KC..MATHia.write expression, positive slope-1                               ***
## KC..MATHia.define variable-1:Opportunity..MATHia.                           ***
## KC..MATHia.enter given, reading numerals-1:Opportunity..MATHia.             ***
## KC..MATHia.enter given, reading words-1:Opportunity..MATHia.                ***
## KC..MATHia.find y, any form-1:Opportunity..MATHia.                          ***
## KC..MATHia.identifying units-1:Opportunity..MATHia.                         ***
## KC..MATHia.write expression, negative intercept-1:Opportunity..MATHia. ***
## KC..MATHia.write expression, negative slope-1:Opportunity..MATHia.      ***
## KC..MATHia.write expression, positive intercept-1:Opportunity..MATHia. ***
## KC..MATHia.write expression, positive slope-1:Opportunity..MATHia.      ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## optimizer (Nelder_Mead) convergence code: 0 (OK)
## Model failed to converge with max|grad| = 0.187458 (tol = 0.002, component 1)
## Model is nearly unidentifiable: very large eigenvalue
##  - Rescale variables?
```

Model without mixed effects

```
b_student_clean$predicted = predict(fit.glm)
error_rate = b_student_clean %>%
  group_by(Opportunity..MATHia.) %>%
  summarise(error = 1 - mean(predicted), actual = 1 - sum(First.Attempt)/n())
ggplot(error_rate, aes(x = Opportunity..MATHia.)) + geom_line(aes(y = error)) +
  geom_line(aes(y = actual, color = "red"))
```


```
summary(fit.glm)
```

```
##
## Call:
## glm(formula = First.Attempt ~ KC..MATHia. + KC..MATHia.:Opportunity..MATHia.,
##     data = b_student_clean)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0023  -0.4855   0.1082   0.3499   0.5431
##
## Coefficients:
##                                                                  Estimate
## (Intercept)                                                     0.8426513
## KC..MATHia.enter given, reading numerals-1                      0.0506151
## KC..MATHia.enter given, reading words-1                        -0.0011343
## KC..MATHia.find y, any form-1                                  -0.3806888
## KC..MATHia.identifying units-1                                 -0.2591566
## KC..MATHia.write expression, negative intercept-1              -0.4070181
## KC..MATHia.write expression, negative slope-1                  -0.4049477
## KC..MATHia.write expression, positive intercept-1              -0.3615797
## KC..MATHia.write expression, positive slope-1                  -0.2942443
## KC..MATHia.define variable-1:Opportunity..MATHia.               0.0061393
## KC..MATHia.enter given, reading numerals-1:Opportunity..MATHia. 0.0023538
## KC..MATHia.enter given, reading words-1:Opportunity..MATHia.    0.0026562
## KC..MATHia.find y, any form-1:Opportunity..MATHia.              0.0047057
## KC..MATHia.identifying units-1:Opportunity..MATHia.             0.0083209
```

```
## KC..MATHia.write expression, negative intercept-1:Opportunity..MATHia.  0.0415420
## KC..MATHia.write expression, negative slope-1:Opportunity..MATHia.      0.0191918
## KC..MATHia.write expression, positive intercept-1:Opportunity..MATHia.  0.0024233
## KC..MATHia.write expression, positive slope-1:Opportunity..MATHia.     -0.0068489
##                                                                        Std. Error
## (Intercept)                                                             0.0089892
## KC..MATHia.enter given, reading numerals-1                              0.0124863
## KC..MATHia.enter given, reading words-1                                 0.0127848
## KC..MATHia.find y, any form-1                                           0.0109366
## KC..MATHia.identifying units-1                                          0.0119733
## KC..MATHia.write expression, negative intercept-1                       0.0236176
## KC..MATHia.write expression, negative slope-1                           0.0208794
## KC..MATHia.write expression, positive intercept-1                       0.0186770
## KC..MATHia.write expression, positive slope-1                           0.0208881
## KC..MATHia.define variable-1:Opportunity..MATHia.                       0.0008464
## KC..MATHia.enter given, reading numerals-1:Opportunity..MATHia.         0.0007608
## KC..MATHia.enter given, reading words-1:Opportunity..MATHia.            0.0008689
## KC..MATHia.find y, any form-1:Opportunity..MATHia.                      0.0002988
## KC..MATHia.identifying units-1:Opportunity..MATHia.                     0.0005903
## KC..MATHia.write expression, negative intercept-1:Opportunity..MATHia.  0.0077737
## KC..MATHia.write expression, negative slope-1:Opportunity..MATHia.      0.0054910
## KC..MATHia.write expression, positive intercept-1:Opportunity..MATHia.  0.0039117
## KC..MATHia.write expression, positive slope-1:Opportunity..MATHia.      0.0054401
##                                                                        t value
## (Intercept)                                                             93.740
## KC..MATHia.enter given, reading numerals-1                               4.054
## KC..MATHia.enter given, reading words-1                                 -0.089
## KC..MATHia.find y, any form-1                                          -34.809
## KC..MATHia.identifying units-1                                         -21.645
## KC..MATHia.write expression, negative intercept-1                      -17.234
## KC..MATHia.write expression, negative slope-1                          -19.395
## KC..MATHia.write expression, positive intercept-1                      -19.360
## KC..MATHia.write expression, positive slope-1                          -14.087
## KC..MATHia.define variable-1:Opportunity..MATHia.                        7.254
## KC..MATHia.enter given, reading numerals-1:Opportunity..MATHia.          3.094
## KC..MATHia.enter given, reading words-1:Opportunity..MATHia.             3.057
## KC..MATHia.find y, any form-1:Opportunity..MATHia.                      15.747
## KC..MATHia.identifying units-1:Opportunity..MATHia.                     14.095
## KC..MATHia.write expression, negative intercept-1:Opportunity..MATHia.   5.344
## KC..MATHia.write expression, negative slope-1:Opportunity..MATHia.       3.495
## KC..MATHia.write expression, positive intercept-1:Opportunity..MATHia.   0.620
## KC..MATHia.write expression, positive slope-1:Opportunity..MATHia.      -1.259
##                                                                        Pr(>|t|)
## (Intercept)                                                             < 2e-16
## KC..MATHia.enter given, reading numerals-1                             5.05e-05
## KC..MATHia.enter given, reading words-1                                0.929302
## KC..MATHia.find y, any form-1                                           < 2e-16
## KC..MATHia.identifying units-1                                          < 2e-16
## KC..MATHia.write expression, negative intercept-1                       < 2e-16
## KC..MATHia.write expression, negative slope-1                           < 2e-16
## KC..MATHia.write expression, positive intercept-1                       < 2e-16
## KC..MATHia.write expression, positive slope-1                           < 2e-16
## KC..MATHia.define variable-1:Opportunity..MATHia.                      4.12e-13
## KC..MATHia.enter given, reading numerals-1:Opportunity..MATHia.        0.001976
```

```
## KC..MATHia.enter given, reading words-1:Opportunity..MATHia.           0.002238
## KC..MATHia.find y, any form-1:Opportunity..MATHia.                      < 2e-16
## KC..MATHia.identifying units-1:Opportunity..MATHia.                     < 2e-16
## KC..MATHia.write expression, negative intercept-1:Opportunity..MATHia. 9.14e-08
## KC..MATHia.write expression, negative slope-1:Opportunity..MATHia.     0.000474
## KC..MATHia.write expression, positive intercept-1:Opportunity..MATHia. 0.535584
## KC..MATHia.write expression, positive slope-1:Opportunity..MATHia.     0.208045
##
## (Intercept)                                                            ***
## KC..MATHia.enter given, reading numerals-1                             ***
## KC..MATHia.enter given, reading words-1
## KC..MATHia.find y, any form-1                                          ***
## KC..MATHia.identifying units-1                                         ***
## KC..MATHia.write expression, negative intercept-1                      ***
## KC..MATHia.write expression, negative slope-1                          ***
## KC..MATHia.write expression, positive intercept-1                      ***
## KC..MATHia.write expression, positive slope-1                          ***
## KC..MATHia.define variable-1:Opportunity..MATHia.                      ***
## KC..MATHia.enter given, reading numerals-1:Opportunity..MATHia.        **
## KC..MATHia.enter given, reading words-1:Opportunity..MATHia.           **
## KC..MATHia.find y, any form-1:Opportunity..MATHia.                     ***
## KC..MATHia.identifying units-1:Opportunity..MATHia.                    ***
## KC..MATHia.write expression, negative intercept-1:Opportunity..MATHia. ***
## KC..MATHia.write expression, negative slope-1:Opportunity..MATHia.     ***
## KC..MATHia.write expression, positive intercept-1:Opportunity..MATHia.
## KC..MATHia.write expression, positive slope-1:Opportunity..MATHia.
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.1782932)
##
##     Null deviance: 10276.4  on 49455  degrees of freedom
## Residual deviance:  8814.5  on 49438  degrees of freedom
## AIC: 55092
##
## Number of Fisher Scoring iterations: 2
```

Two-Sample T-Test - Retrospective Study

```
library(tidyverse)
library(ggplot2)
library(lme4)
library(dplyr)
```

Read in data and pairs of correlated workspaces from Gaussian Graphical Model

```
df = read.delim("course2_1920_sample1_500students_datashop.txt", header = T)
t = read.csv("df_success_rate.csv")
```

A function that calculates p-value of the t-test and the coefficient of a linear model that regresses treatment on success rate.

```
get_p_value = function(ws.1, ws.2){
  #filter and get all data from workspace 1 and workspace 2
  temp = df[which(df$Level..Workspace.Id. %in% c(ws.1, ws.2)), ]
  #get start time of each workspace of every student
  temp.time = temp %>%
    group_by(Anon.Student.Id, Level..Workspace.Id.) %>%
    summarise(start.time = min(Time))

  #add labels of workspace order based on time
  result = NA
  for (id in unique(temp.time$Anon.Student.Id)){
    t.1 = temp.time[which(temp.time$Anon.Student.Id == id &
                            temp.time$Level..Workspace.Id. == ws.1), ]
    t.2 = temp.time[which(temp.time$Anon.Student.Id == id &
                            temp.time$Level..Workspace.Id. == ws.2), ]
    if (nrow(t.1) == 0){
      label = "only ws 2"
    } else if (nrow(t.2) == 0){
      label = "only ws 1"
    } else if (t.1$start.time < t.2$start.time){
      label = "ws 1 first"
    } else {
      label = "ws 2 first"
    }
    if (is.na(result)){
      result = data.frame("id" = id, "label" = label)
    } else{
      result = rbind(result, c(id, label))
    }
  }

  #we only care about these three cases in the experiment
  id.v = result$id[which(result$label %in% c("ws 1 first", "ws 2 first", "only ws 2"))]

  #we are interested in student's performance on workspace 2
  #we look at success rate of the first attempy
  test = df[which(df$Anon.Student.Id %in% id.v & df$Attempt.At.Step == 1
                  & df$Level..Workspace.Id. == ws.2), ]
  success = test %>%
```

```
    group_by(Anon.Student.Id) %>%
    summarise(success.rate = sum(Outcome == "OK")/sum(Attempt.At.Step))
  names(success) = c("id", "success.rate")

  #join success rate table with label table
  bozo = full_join(success, result, by = "id")

  #add indicator variable for treatment - workspace 1
  bozo$ind = ifelse(bozo$label == "ws 1 first", 1, 0)
  bozo = na.omit(bozo)

  #In cases without natural experiment, we return text
  if (length(which(bozo$ind == 1)) <= 1){
    return("no ws1 first")
  }
  if (length(which(bozo$ind == 0)) <= 1){
    return("no ws2 first + only ws2")
  }

  #get t test p-value
  p = t.test(bozo[which(bozo$ind == 1), 2], bozo[which(bozo$ind == 0), 2],
             alternative = "greater")

  #get linera regression coefficient
  fit = lm(success.rate ~ ind, data = bozo)
  effect = summary(fit)$coef[[2]]

  return(c(p[[3]], effect))
}
```

Loop through all pairs of workspaces in the csv file and get results of p-values and effects

```
p.list = c()
effect.list = c()
p.list.2 = c()
effect.list.2 = c()

for (i in 1:nrow(t)){
  ws.1 = t[i, 2]
  ws.2 = t[i, 4]
  result = get_p_value(ws.1, ws.2)
  if (length(result) == 1){
    p.list = c(p.list, result)
    effect.list = c(effect.list, "")
  } else{
    p.list = c(p.list, result[1])
    effect.list = c(effect.list, result[2])
  }

  ws.2 = t[i, 2]
  ws.1 = t[i, 4]
  result = get_p_value(ws.1, ws.2)
  if (length(result) == 1){
    p.list.2 = c(p.list.2, result)
```

```
    effect.list.2 = c(effect.list.2, "")
  } else{
    p.list.2 = c(p.list.2, result[1])
    effect.list.2 = c(effect.list.2, result[2])
  }
}
```

Write results into a csv file

```
dat = cbind(t, p.list)
dat = cbind(dat, effect.list)
dat = cbind(dat, p.list.2)
dat = cbind(dat, effect.list.2)
names(dat) = c("Node_1", "Workspace_1", "Node_2", "Workspace_2", "Weight",
               "P-Value", "Effect", "P-Value 2", "Effect 2")

write.csv(dat, "df_success_rate_result.csv")
```

Appendix 4

In this technical appendix, we work with the MATHia course 2 (Grade 7) dataset of 500 randomly selected students. Using Gaussian Graphical Models, we find out strongly related workspaces and the partial correlation coefficient for each pair. We later using retrospective study design to evaluate the prerequisite relation between the workspaces.

```
head(mathia)
```

```
##                        Anon.Student.Id          Session.Id         Time
## 1 00a7bb8b-a418-4ac5-b252-58c2d9918f62 no_session_tracking 1.566933e+12
## 2 00a7bb8b-a418-4ac5-b252-58c2d9918f62 no_session_tracking 1.566933e+12
## 3 00a7bb8b-a418-4ac5-b252-58c2d9918f62 no_session_tracking 1.566934e+12
## 4 00a7bb8b-a418-4ac5-b252-58c2d9918f62 no_session_tracking 1.566934e+12
## 5 00a7bb8b-a418-4ac5-b252-58c2d9918f62 no_session_tracking 1.566934e+12
## 6 00a7bb8b-a418-4ac5-b252-58c2d9918f62 no_session_tracking 1.566934e+12
##   Level..Workspace.Id.           Problem.Name     Step.Name   Selection  Action
## 1  pre_launch_protocol pre_launch_protocol_01  component6.1             Attempt
## 2  pre_launch_protocol pre_launch_protocol_01  component6.1             Attempt
## 3  pre_launch_protocol pre_launch_protocol_01  component6.1             Attempt
## 4  pre_launch_protocol pre_launch_protocol_01  component8.1             Attempt
## 5  pre_launch_protocol pre_launch_protocol_01 component11.1             Attempt
## 6  pre_launch_protocol pre_launch_protocol_01               Done Button    Done
##                                                          Input Outcome
## 1                                                          320     BUG
## 2                                                          320     BUG
## 3                                                       25,000      OK
## 4 They have a lot of practice remembering a lot of locations.      OK
## 5                                                        grows      OK
## 6                                                                   OK
##   Help.Level Attempt.At.Step KC.Model.MATHia.                     CF..Ruleid.
## 1          0               1                  Student entered first JIT answer
## 2          0               1                  Student entered first JIT answer
## 3          0               2
## 4          0               1
## 5          0               1
## 6          0               0
##                                                 CF..Etalon.
## 1                                                    25,000
## 2                                                    25,000
## 3                                                    25,000
## 4 They have a lot of practice remembering a lot of locations.
## 5                                                     grows
## 6
##   CF..Skill.Previous.p.Known. CF..Skill.New.p.Known.
## 1                          NA                    NA
## 2                          NA                    NA
## 3                          NA                    NA
## 4                          NA                    NA
```

```
## 5                          NA                    NA
## 6                          NA                    NA
##   CF..Workspace.Progress.Status.              CF..Semantic.Event.Id.
## 1                      GRADUATED 9bfa9505-b4fc-4576-84a8-04adbccffc3d
## 2                      GRADUATED 9bfa9505-b4fc-4576-84a8-04adbccffc3d
## 3                      GRADUATED 230e2c5c-efc2-4d45-ae28-a62ee4696380
## 4                      GRADUATED bc6b902e-3577-47cb-a937-08fa9e580df8
## 5                      GRADUATED 04365e3e-d44d-4b92-a4ea-75dded404187
## 6                      GRADUATED 5d151ef3-81a5-4695-94bc-4e21e0a79871
```

```
head(imp_workspaces)
```

```
## # A tibble: 6 x 1
##   `TOC for SequenceTemplate : Middle School Math Solution Course 2 ( msms_cours~
##   <chr>
## 1 A1 : Pre-Launch Protocol  (pre_launch_protocol)
## 2 U : Pre-Launch Protocol
## 3 S : Pre-Launch Protocol  (pre_launch_protocol)
## 4 A2 : Thinking Proportionally  (msms_course2_module1)
## 5 U : Circles
## 6 S : Investigating Circles  (investigating_circles)
```

Getting workspaces from MATHia 'shipped' Course 2 curriculum.

```
imp_workspaces2 <- imp_workspaces %>%
  filter(substr(`TOC for SequenceTemplate : Middle School Math Solution Course 2 ( msms_course2 )`,
                1, 1) == "S") %>%
  mutate(Start = str_locate(`TOC for SequenceTemplate : Middle School Math Solution Course 2 ( msms_cou
                            "\\("),
         End = nchar(`TOC for SequenceTemplate : Middle School Math Solution Course 2 ( msms_course2 )`)
  mutate(wks = substr(`TOC for SequenceTemplate : Middle School Math Solution Course 2 ( msms_course2 )
                      Start+1, End-1))
imp_workspaces3 <- imp_workspaces2 %>%
  select(wks) %>%
  mutate(IF = str_locate(wks, "\\("))
imp_workspaces3$IF <- ifelse(is.na(imp_workspaces3$IF)==TRUE, 0,
                             imp_workspaces3$IF)
imp_workspaces4 <- imp_workspaces3 %>%
  mutate(workspaces = substr(wks, IF+1, length(wks))) %>%
  select(workspaces) %>%
  filter(!(workspaces %in% c("pre_launch_protocol", "factoring_expression",
                             "scale_drawings_", "understanding_opposite"))) %>%
  add_row(workspaces = c("factoring_expressions", "scale_drawings_3",
                         "understanding_opposites"))
```

The workspaces from MATHia Course 2 'shipped' curriculum.

```
sort(imp_workspaces4$workspaces)
```

```
##  [1] "adding_and_subtracting_integers"
##  [2] "analyzing_different_forms_of_expressions"
##  [3] "analyzing_models_2step_integers"
##  [4] "area_perimeter_circle_forwards"
##  [5] "calculating_angles"
```

```
##  [6] "calculating_compound_probabilities"
##  [7] "classify_and_determine_angles"
##  [8] "comparing_characteristics_of_data_displays"
##  [9] "comparing_populations_using_data_displays"
## [10] "comparing_theoretical_and_experimental_probabilities"
## [11] "conceptual_equations_2step_1var"
## [12] "converting_rational_numbers_to_decimals"
## [13] "converting_with_fractional_percents"
## [14] "critical_attributes_of_similar_figures"
## [15] "determining_characteristics_of_direct_variation_graphs"
## [16] "determining_probabilities"
## [17] "determining_the_value_of_an_independent_variable"
## [18] "developing_algorithms_for_adding_and_subtracting_integers"
## [19] "direct_variation_convert"
## [20] "direct_variation_equation"
## [21] "distinguishing_between_populations_and_samples"
## [22] "exploring_proportions_and_direct_variation"
## [23] "factoring_expressions"
## [24] "fractional_percent_models"
## [25] "fractional_rates"
## [26] "graphs_of_equations"
## [27] "identifying_signs_of_starting_values_and_rates"
## [28] "integer_add_subtract_nl_mix"
## [29] "introduction_to_compound_events"
## [30] "investigating_circles"
## [31] "linear_inequalities_numberline"
## [32] "linear_inequalities_solver_1step"
## [33] "linear_inequalities_solver_2step"
## [34] "linear_relations_1"
## [35] "multiplying_and_dividing_integers"
## [36] "picture_algebra_mix_variable"
## [37] "proportional_relationships"
## [38] "ratio_proportion_change3"
## [39] "ratio_proportion_change4"
## [40] "ratio_proportion_prop1"
## [41] "ratio_proportion_prop2"
## [42] "ratio_proportion_ratio2"
## [43] "rewriting_proportions_as_products"
## [44] "sales_tax_discounts_one_rate"
## [45] "sales_tax_discounts_two_rates"
## [46] "scale_drawings_3"
## [47] "simplify_order_of_ops_expression_numeric_contrast_addsub_multdiv"
## [48] "simplify_order_of_ops_expression_numeric_mix_type_complex"
## [49] "simplify_order_of_ops_expression_numeric_mix_type_simple"
## [50] "simplify_order_of_ops_expression_numeric_parens_expon"
## [51] "simplify_order_of_ops_expression_variable_complex"
## [52] "simplify_order_of_ops_expression_variable_contrast_addsub_multdiv"
## [53] "simplify_order_of_ops_expression_variable_four_ops_complex"
## [54] "simplify_order_of_ops_expression_variable_mix_type"
## [55] "simplify_order_of_ops_expression_variable_parens_expon"
## [56] "simulating_compound_events"
## [57] "simulating_simple_events"
## [58] "solve_linear_equation_2step"
## [59] "solve_linear_equation_2step_div_notype"
```

```
## [60] "solve_linear_equation_2step_div_typein"
## [61] "solve_linear_equation_2step_mult_notype"
## [62] "solve_linear_equation_2step_mult_typein"
## [63] "solving_simple_percent_problems"
## [64] "understanding_opposites"
## [65] "understanding_volume_of_right_prisms"
## [66] "using_graphs_to_solve_equations"
## [67] "using_proportions_to_solve_percent_problems"
## [68] "using_scale_drawings"
## [69] "visualizing_cross_sections_of_three_dimensional_shapes"
## [70] "volume_surface_area_right_prism_vol-_backward"
## [71] "volume_surface_area_sq_pyramid_vol"
## [72] "volume_surface_area_sq_pyramid_vol-_backward"
## [73] "worksheet_grapher_a1_direct_variation"
## [74] "worksheet_grapher_a1_patterns_2step_expr"
## [75] "worksheet_grapher_a1_solving_2step_dec_frac"
## [76] "worksheet_grapher_a1_solving_2step_int"
```

Filtering workspaces from the ones in MATHia Course 2 'shipped' curriculum.

```
mathia2 <- mathia %>%
  filter(Level..Workspace.Id. %in% imp_workspaces4$workspaces)
```

```
# Sanity check
imp_workspaces4$workspaces[!imp_workspaces4$workspaces %in%
                             unique(mathia$Level..Workspace.Id.)]
```

```
## character(0)
```

```
length(unique(mathia2$Level..Workspace.Id.))
```

```
## [1] 76
```

We want to find success rate for first attempts of all steps in a workspace. So filtering by `Attempt.At.Step == 1`.

```
mathia3 <- mathia2 %>%
  group_by(Anon.Student.Id, Level..Workspace.Id., Problem.Name, Step.Name) %>%
  filter(Attempt.At.Step == 1) %>%
  select(Anon.Student.Id, Level..Workspace.Id., Problem.Name, Step.Name,
         Outcome, Attempt.At.Step)
```

Converting `Outcome` to success, its numeric version

```
mathia4 <- mathia3 %>%
  mutate(success = ifelse(Outcome == "OK", 1, 0)) %>%
  select(-Attempt.At.Step)
```

Calculating mean success rate as `success_rate`.

```
mathia5 <- mathia4 %>%
  ungroup() %>%
  group_by(Anon.Student.Id, Level..Workspace.Id.) %>%
  summarise(success_rate = mean(success))
```

Transforming the data frame using `pivot_wider` to be able to compute correlations.

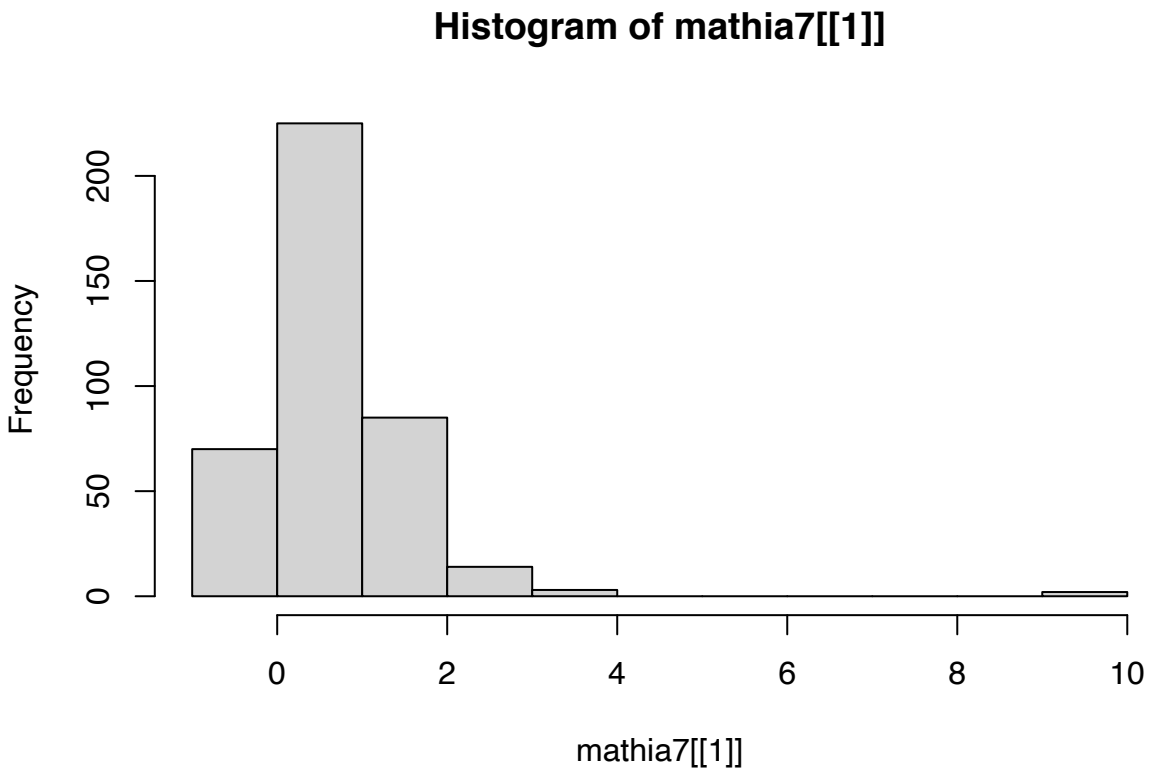
```
mathia6 <- mathia5 %>%
  ungroup() %>%
  pivot_wider(names_from = Level..Workspace.Id.,
              values_from = success_rate) %>%
  select(-Anon.Student.Id)
```

Applying logit function.

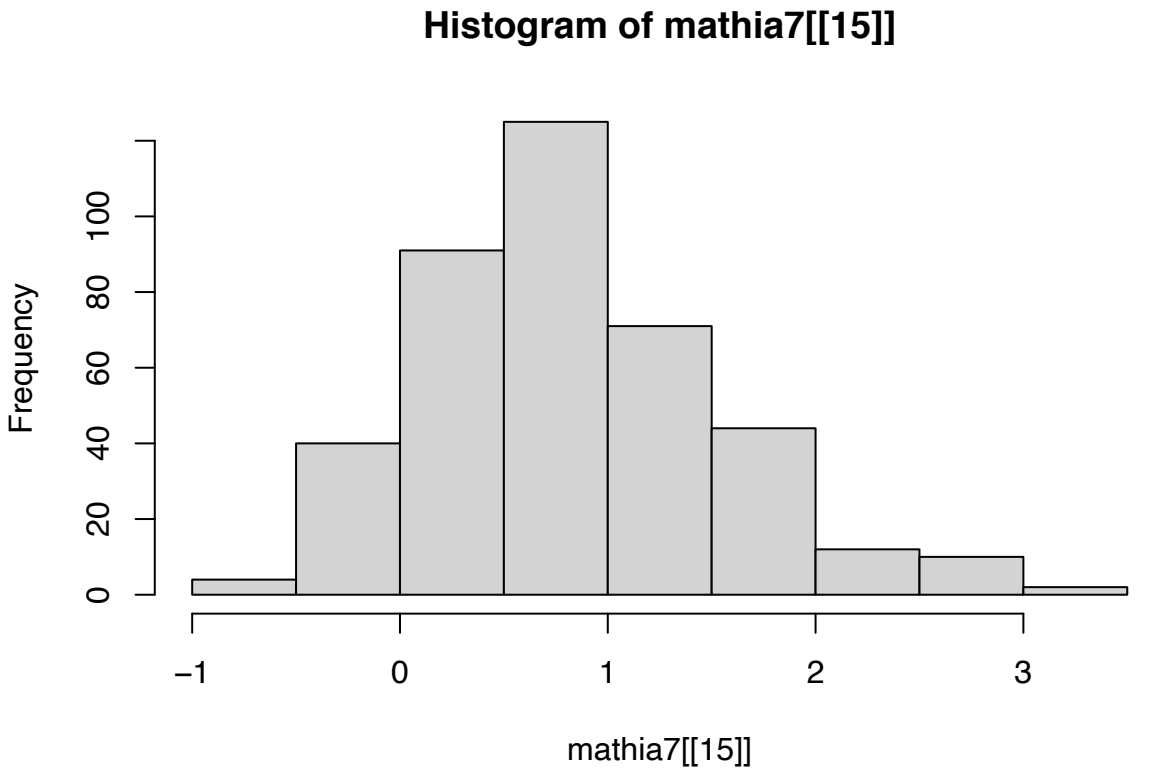
```
mathia7 <- mathia6
logit <- function(v){
  if (is.na(v)==TRUE){
    return(NA)
  }
  if (is.nan(v) == TRUE){
    return(NA)
  }
  if (v==1){
    v = 0.9999
  }
  if (v==0){
    v = 0.0001
  }
  return(log(v/(1-v)))
}
for (i in colnames(mathia7)){
  mathia7[[i]] <- lapply(mathia7[[i]], logit)
}
for (i in colnames(mathia7)){
  mathia7[[i]] <- as.numeric(mathia7[[i]])
}
```

Validating the normality assumption for some workspaces.

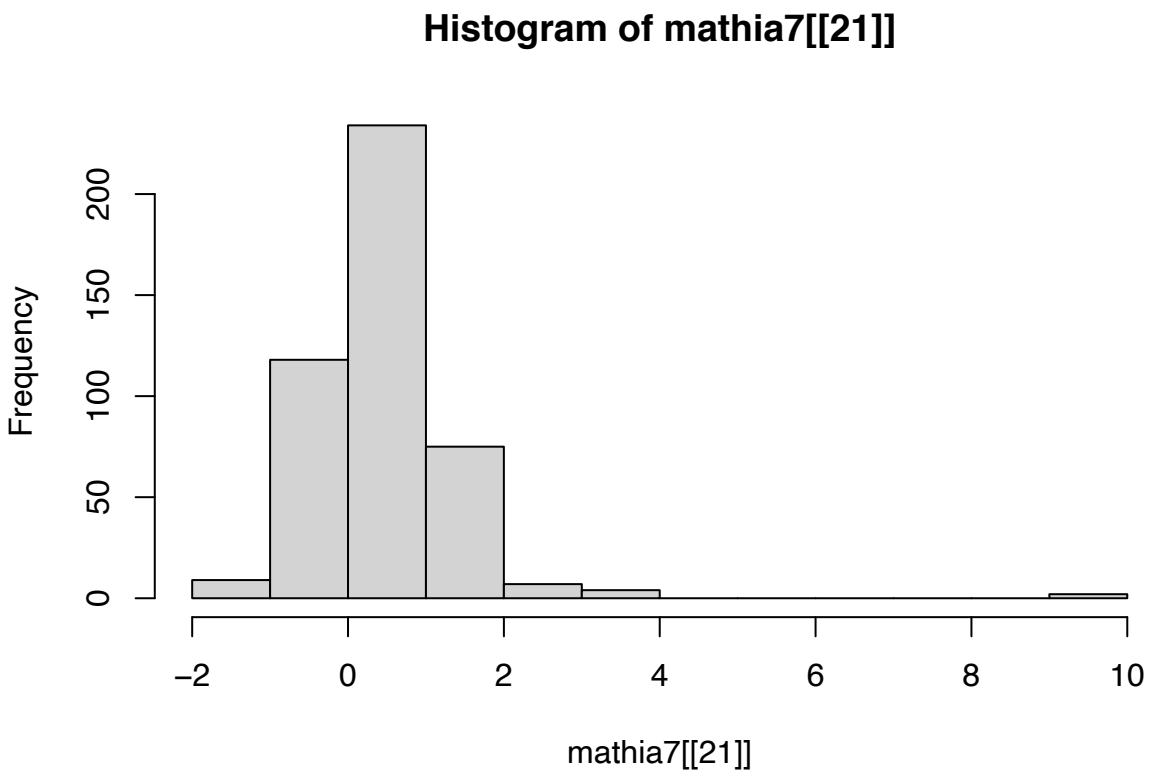
```
hist(mathia7[[1]])
```



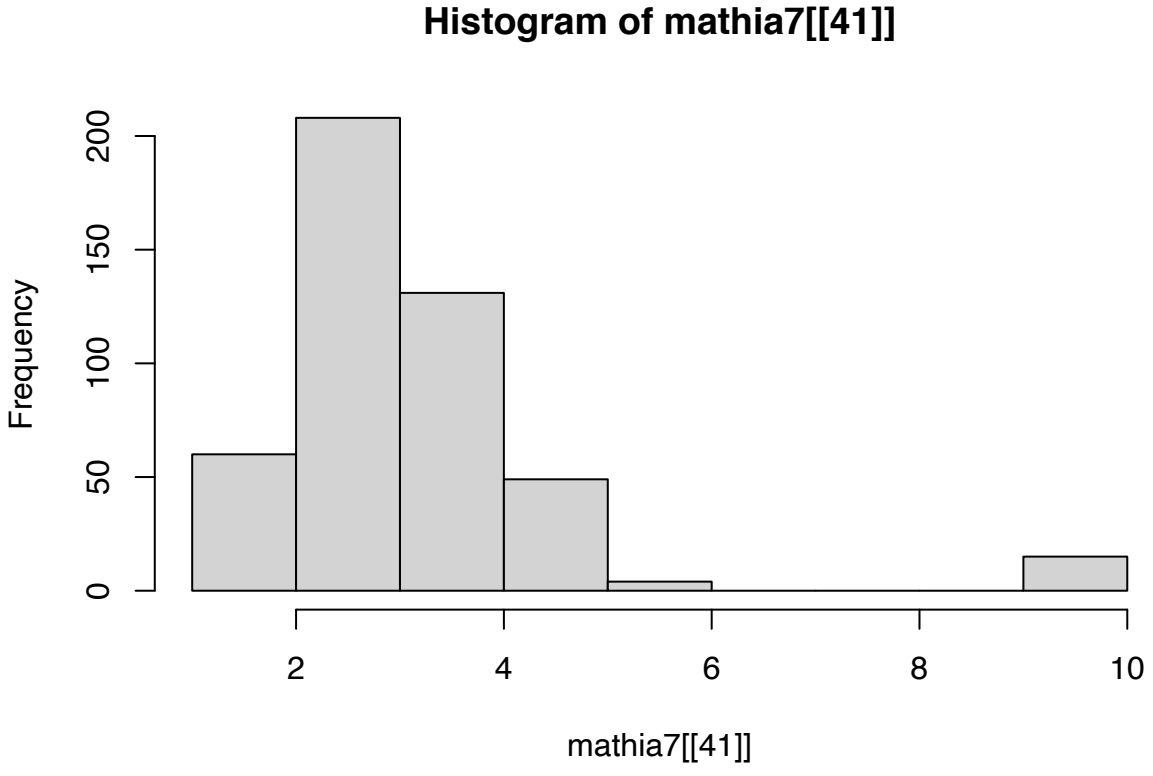

```
hist(mathia7[[15]])
```



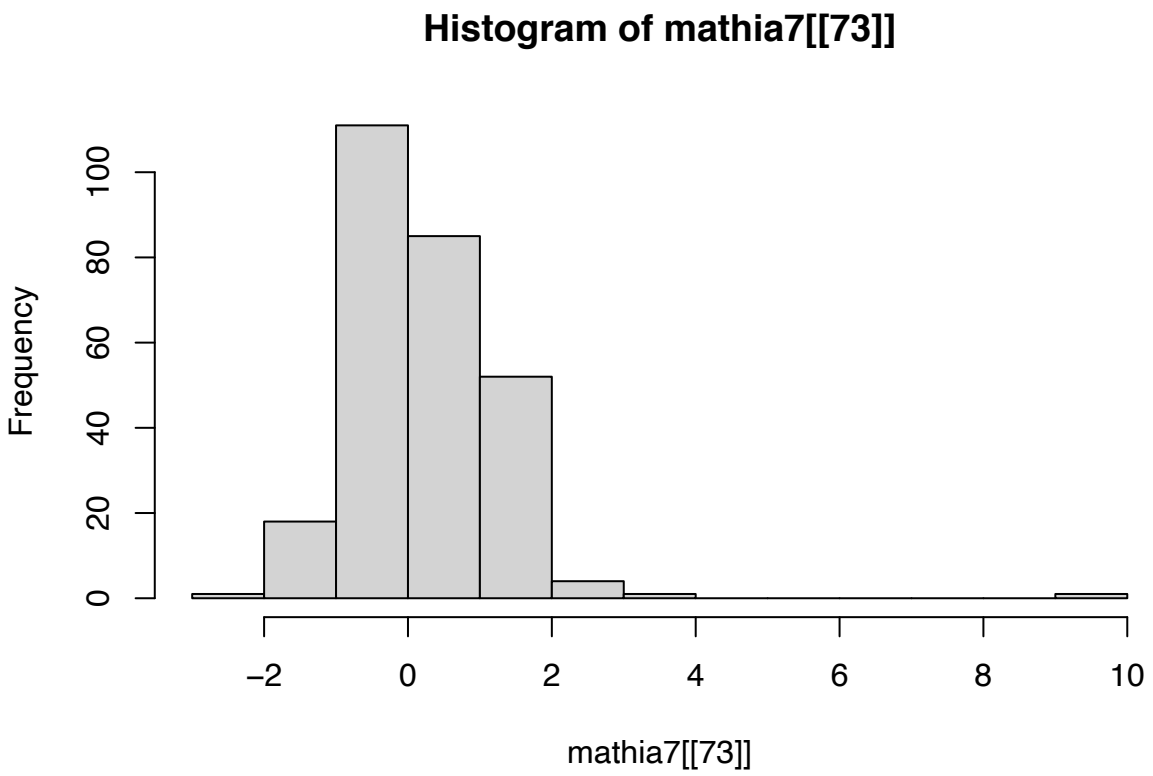

```
hist(mathia7[[21]])
```




```
hist(mathia7[[41]])
```




```
hist(mathia7[[73]])
```

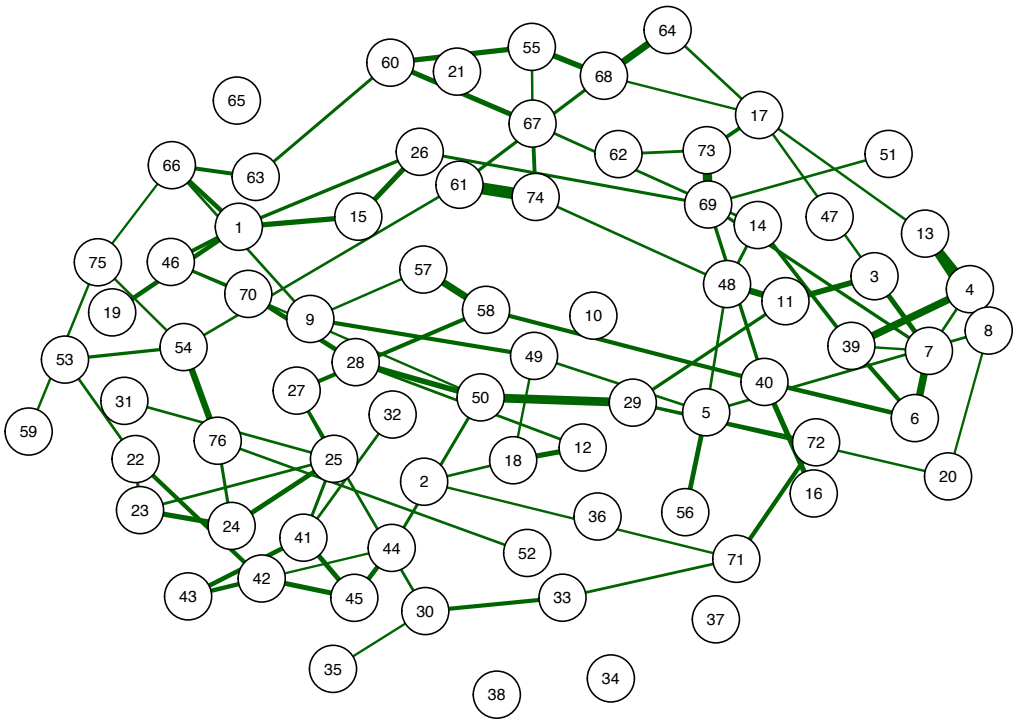


Computing pairwise correlations and converting the matrix to positive definite.

```
# mathia_cor_pd <- psych::corFiml(mathia7) gives error
mathia_cor_success_rate_pairwise <- cor(mathia7, use = "pairwise.complete.obs")
mathia_cor_success_rate <- psych::cor.smooth(mathia_cor_success_rate_pairwise)
```

GGM plot where student performance metric is success rate of first attempts for all steps in a workspace.

```
ggm_success_rate <- qgraph::qgraph(
      mathia_cor_success_rate,
      layout = "spring",
      graph = "glasso",
      labels = TRUE,
      minimum = 0.1,
      sampleSize = 500
    )
```



Getting the pairs of related workspaces from GGM along with partial correlation coefficients.

```
workspaces_data_success_rate <- data.frame(Node = 1 : 76,
                            Workspace = colnames(mathia_cor_success_rate))
```

```
df_success_rate <- data.frame(Node_1 = ggm_success_rate$Edgelist$from,
                            Node_2 = ggm_success_rate$Edgelist$to,
                            Weight = ggm_success_rate$Edgelist$weight) %>%
  arrange(desc(Weight)) %>%
  full_join(workspaces_data_success_rate, by = c("Node_1" = "Node")) %>%
  rename(Workspace_1 = Workspace) %>%
  full_join(workspaces_data_success_rate, by = c("Node_2" = "Node")) %>%
  rename(Workspace_2 = Workspace) %>%
  relocate(Node_1, Workspace_1, Node_2, Workspace_2, Weight)
```

```
head(df_success_rate)
```

```
##   Node_1                                        Workspace_1 Node_2
## 1     61                  solving_simple_percent_problems     74
## 2     69                               graphs_of_equations     73
## 3     29                                  scale_drawings_3     50
## 4      8 comparing_theoretical_and_experimental_probabilities     13
## 5      4                 calculating_compound_probabilities     13
## 6      4                 calculating_compound_probabilities     39
##                                  Workspace_2    Weight
## 1   using_proportions_to_solve_percent_problems 0.4122318
## 2               using_graphs_to_solve_equations 0.3292839
## 3 volume_surface_area_right_prism_vol-_backward 0.3116518
## 4                    determining_probabilities 0.2874725
## 5                    determining_probabilities 0.2742894
## 6                   simulating_compound_events 0.2701656
```

Now, the student performance metric is assistance score over a workspace. Calculating assistance score for a student over a workspace.

```
mathia8 <- mathia2 %>%
  group_by(Anon.Student.Id, Level..Workspace.Id.) %>%
  select(Anon.Student.Id, Level..Workspace.Id., Outcome)
```

```
mathia9 <- mathia8 %>%
  mutate(assistance = ifelse(Outcome == "OK", 0, 1)) %>%
  select(-Outcome)
```

```
mathia10 <- mathia9 %>%
  summarise(assistance_score = sum(assistance))
```

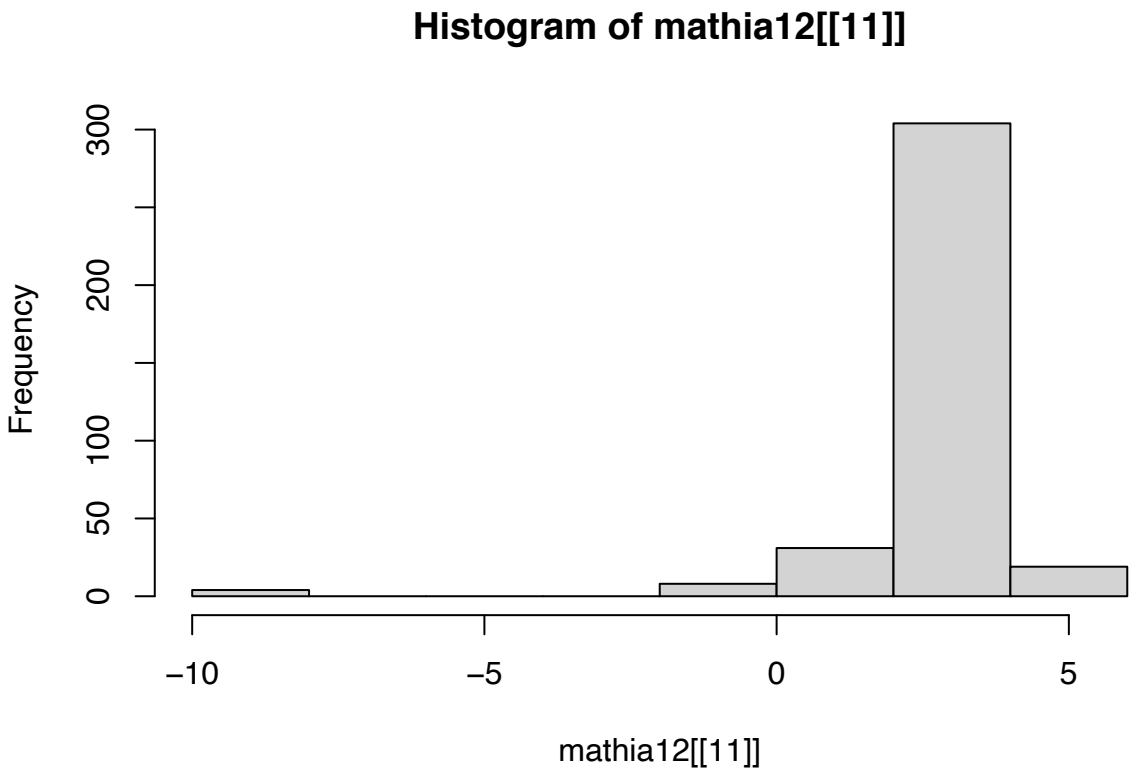
Transforming the data frame using `pivot_wider` to be able to compute correlations.

```
mathia11 <- mathia10 %>%
  ungroup() %>%
  pivot_wider(names_from = Level..Workspace.Id.,
              values_from = assistance_score) %>%
  select(-Anon.Student.Id)
```

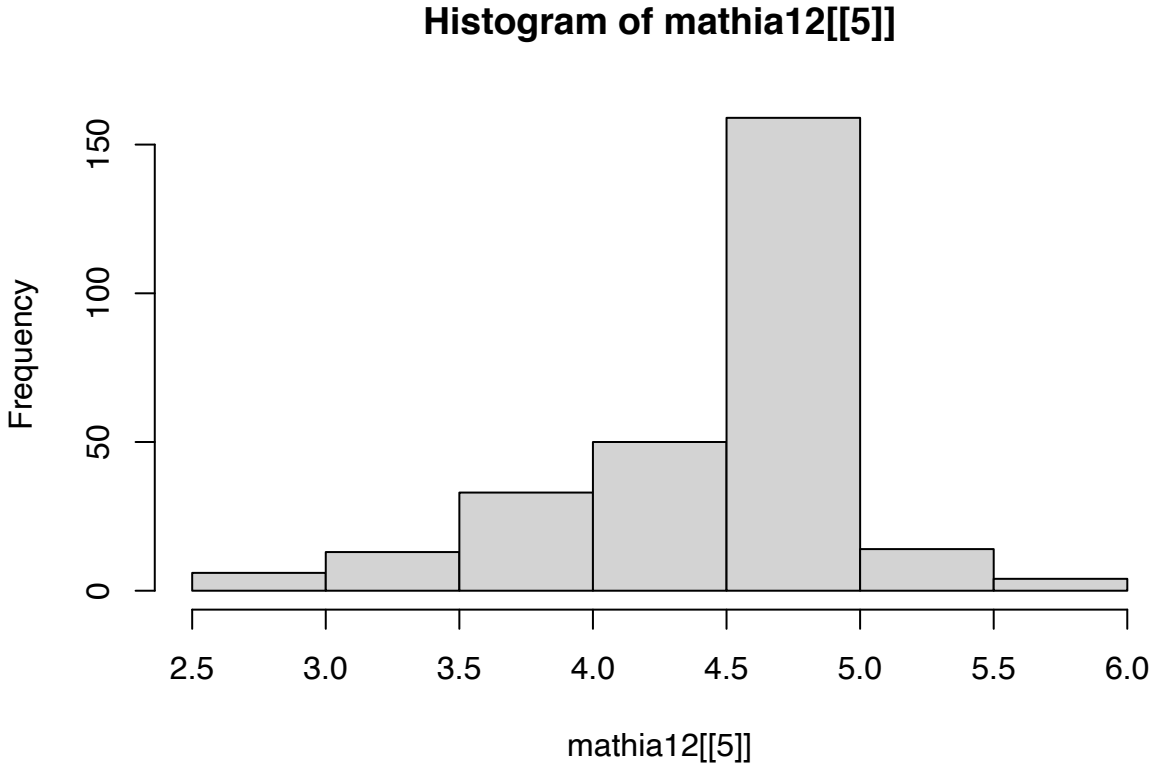
```
mathia12 <- mathia11
log2 <- function(num){
  if (is.na(num)==TRUE || is.null(num)==TRUE){
    return(NA)
  }
  if (num == 0){
    return(log(0.0001))
  }
  return(log(num))
}
for (i in colnames(mathia12)){
  mathia12[[i]] <- lapply(mathia12[[i]], log2)
}
for (i in colnames(mathia12)){
  mathia12[[i]] <- as.numeric(mathia12[[i]])
}
```

Validating the normality assumption for some workspaces.

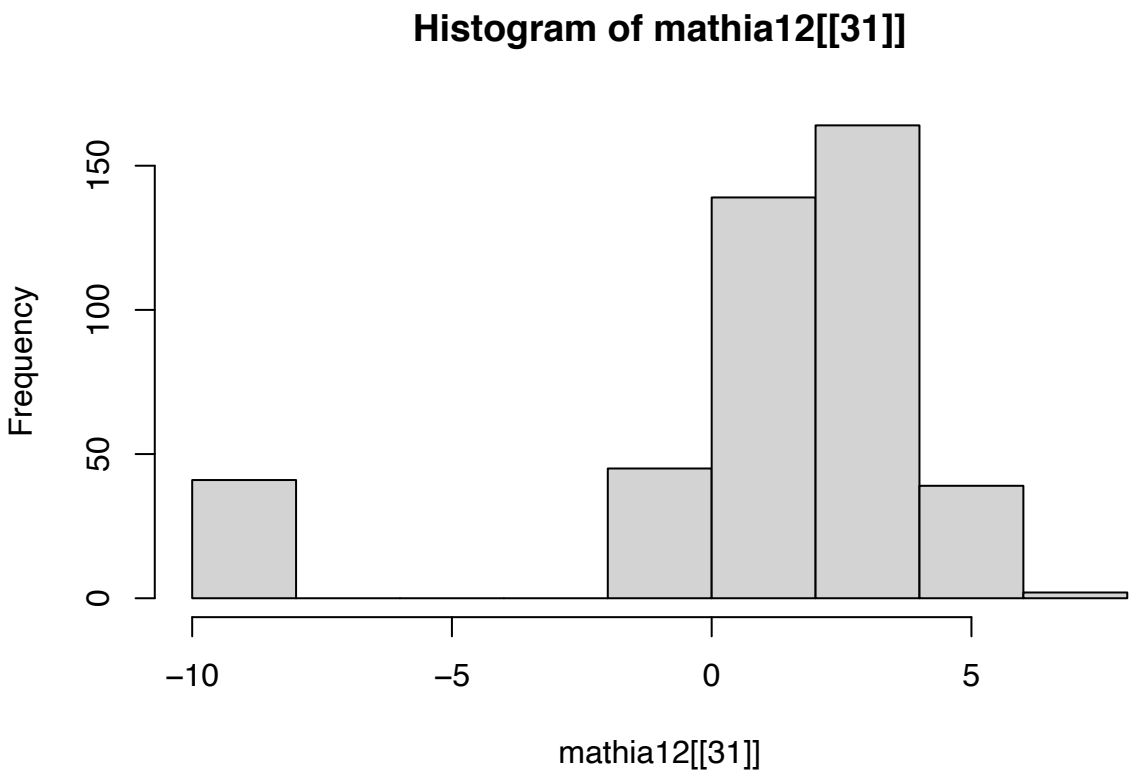
```
hist(mathia12[[11]])
```



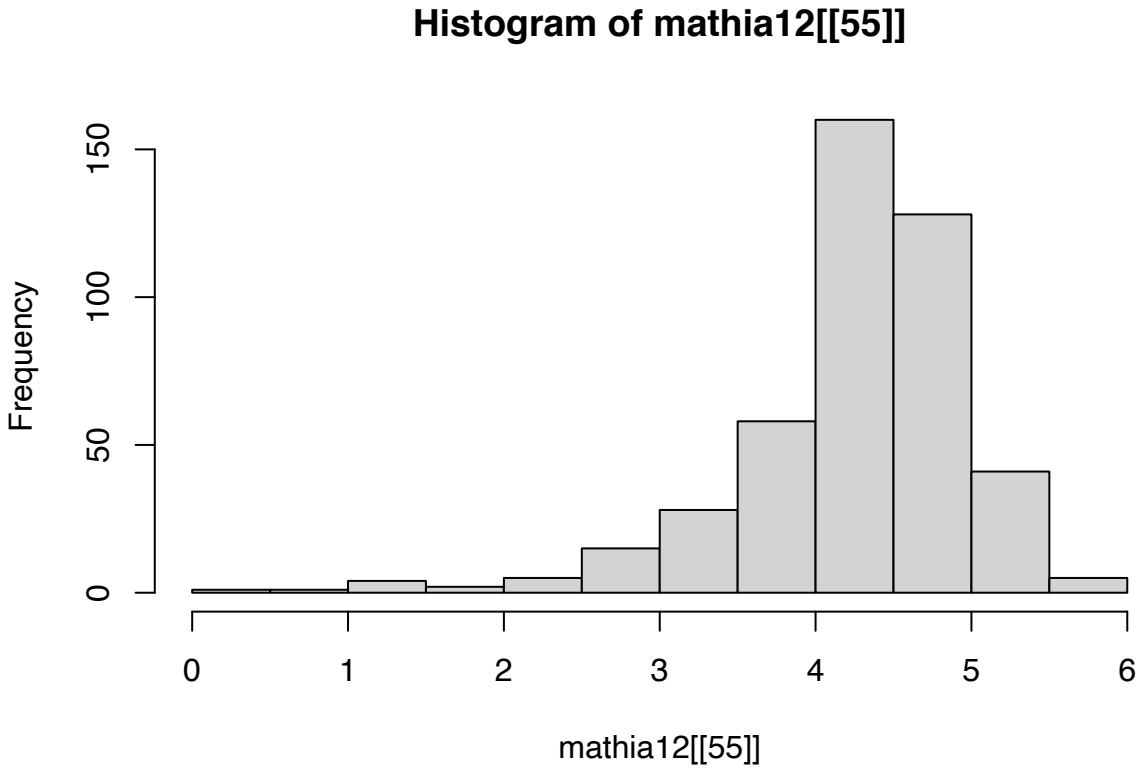

```
hist(mathia12[[5]])
```



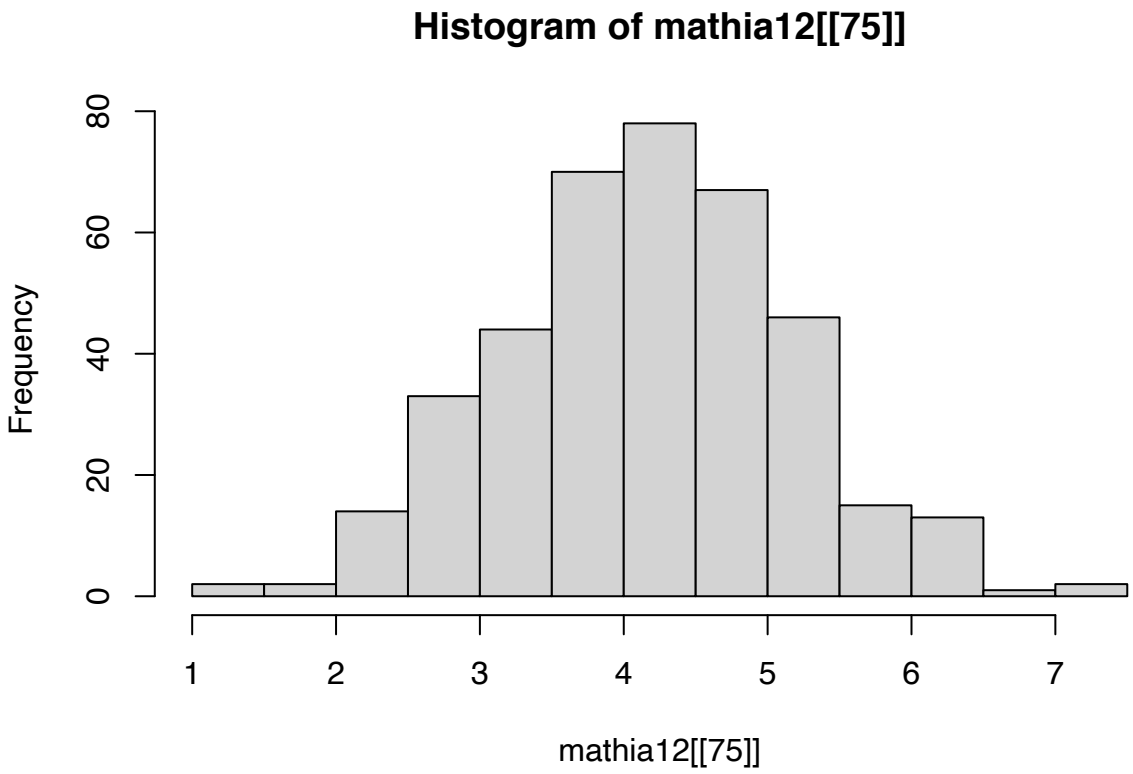

```
hist(mathia12[[31]])
```




```
hist(mathia12[[55]])
```




```
hist(mathia12[[75]])
```

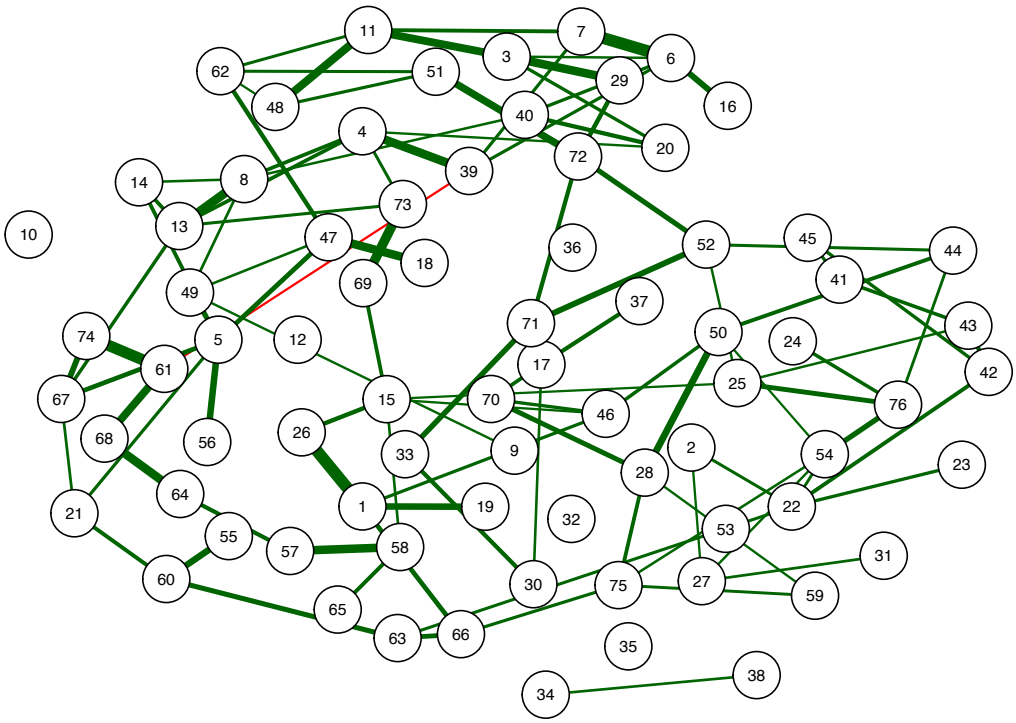


Computing pairwise correlations and converting the matrix to positive definite.

```
# mathia_cor_pd <- psych::corFiml(mathia12) gives error
mathia_cor_assistance_score_pairwise <- cor(mathia12,
                                            use = "pairwise.complete.obs")
mathia_cor_assistance_score <-
  psych::cor.smooth(mathia_cor_assistance_score_pairwise)
```

GGM plot where student performance metric is assistance score across a workspace.

```
ggm_assistance_score <- qgraph::qgraph(
      mathia_cor_assistance_score,
      layout = "spring",
      graph = "glasso",
      labels = TRUE,
      minimum = 0.1,
      sampleSize = 500
    )
```



Getting the pairs of related workspaces from GGM along with partial correlation coefficients.

```
workspaces_data_assistance_score <- data.frame(Node = 1 : 76,
                                Workspace = colnames(mathia_cor_assistance_score))
```

```
df_assistance_score <- data.frame(Node_1 = ggm_assistance_score$Edgelist$from,
                                Node_2 = ggm_assistance_score$Edgelist$to,
                                Weight = ggm_assistance_score$Edgelist$weight) %>%
  arrange(desc(Weight)) %>%
  full_join(workspaces_data_assistance_score, by = c("Node_1" = "Node")) %>%
  rename(Workspace_1 = Workspace) %>%
  full_join(workspaces_data_assistance_score, by = c("Node_2" = "Node")) %>%
  rename(Workspace_2 = Workspace) %>%
  relocate(Node_1, Workspace_1, Node_2, Workspace_2, Weight)
```

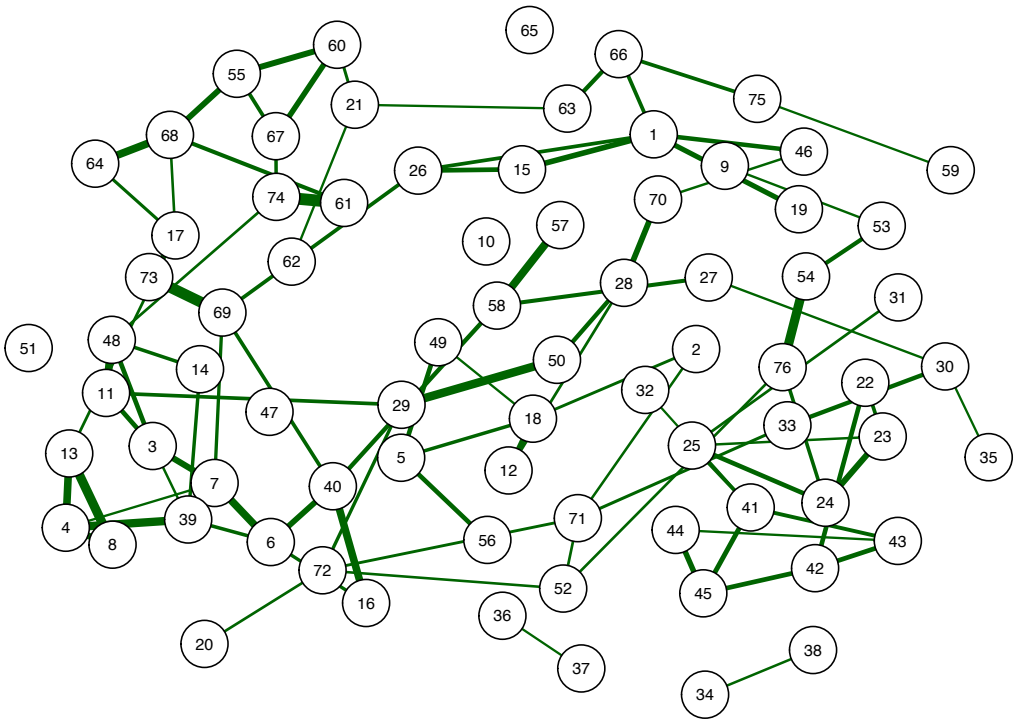
```
head(df_assistance_score)
```

```
##   Node_1                                        Workspace_1 Node_2
## 1      6          comparing_characteristics_of_data_displays      7
## 2      1                      adding_and_subtracting_integers     26
## 3     61                      solving_simple_percent_problems     74
## 4     69                                graphs_of_equations     73
## 5      8 comparing_theoretical_and_experimental_probabilities     13
## 6      4                  calculating_compound_probabilities     39
##                                Workspace_2    Weight
## 1   comparing_populations_using_data_displays 0.3527237
## 2           multiplying_and_dividing_integers 0.3512463
## 3 using_proportions_to_solve_percent_problems 0.3368581
## 4             using_graphs_to_solve_equations 0.2925665
## 5                  determining_probabilities 0.2746823
## 6                 simulating_compound_events 0.2677408
```

Examining the GGM plot when student performance metric is 'first 2 attempts' for all steps and 'first 3 attempts' for all steps in a workspace.

```
mathia13 <- mathia2 %>%
  group_by(Anon.Student.Id, Level..Workspace.Id., Problem.Name, Step.Name) %>%
  filter(Attempt.At.Step %in% c(1, 2)) %>%
  select(Anon.Student.Id, Level..Workspace.Id., Problem.Name, Step.Name,
         Outcome, Attempt.At.Step) %>%
  mutate(success = ifelse(Outcome == "OK", 1, 0)) %>%
  select(-Outcome) %>%
  ungroup() %>%
  group_by(Anon.Student.Id, Level..Workspace.Id., Problem.Name, Step.Name) %>%
  summarise(success_rate1 = mean(success)) %>%
  ungroup() %>%
  group_by(Anon.Student.Id, Level..Workspace.Id.) %>%
  summarise(success_rate = mean(success_rate1)) %>%
  pivot_wider(names_from = Level..Workspace.Id., values_from = success_rate) %>%
  ungroup() %>%
  select(-Anon.Student.Id)
```

```
for (i in colnames(mathia13)){
  mathia13[[i]] <- lapply(mathia13[[i]], logit)
}
for (i in colnames(mathia13)){
  mathia13[[i]] <- as.numeric(mathia13[[i]])
}
mathia_cor_success_rate_pairwise_2attempts <- cor(mathia13,
                                                  use = "pairwise.complete.obs")
mathia_cor_success_rate_2attempts <-
  psych::cor.smooth(mathia_cor_success_rate_pairwise_2attempts)
ggm_success_rate_2attempts <- qgraph::qgraph(
      mathia_cor_success_rate_2attempts,
      layout = "spring",
      graph = "glasso",
      labels = TRUE,
      minimum = 0.1,
      sampleSize = 500
    )
```




```
mathia14 <- mathia2 %>%
  group_by(Anon.Student.Id, Level..Workspace.Id., Problem.Name, Step.Name) %>%
  filter(Attempt.At.Step %in% c(1, 2, 3)) %>%
  select(Anon.Student.Id, Level..Workspace.Id., Problem.Name, Step.Name,
         Outcome, Attempt.At.Step) %>%
  mutate(success = ifelse(Outcome == "OK", 1, 0)) %>%
  select(-Outcome) %>%
  ungroup() %>%
  group_by(Anon.Student.Id, Level..Workspace.Id., Problem.Name, Step.Name) %>%
  summarise(success_rate1 = mean(success)) %>%
  ungroup() %>%
  group_by(Anon.Student.Id, Level..Workspace.Id.) %>%
  summarise(success_rate = mean(success_rate1)) %>%
  pivot_wider(names_from = Level..Workspace.Id., values_from = success_rate) %>%
  ungroup() %>%
  select(-Anon.Student.Id)
```

```
for (i in colnames(mathia14)){
  mathia14[[i]] <- lapply(mathia14[[i]], logit)
}
for (i in colnames(mathia14)){
  mathia14[[i]] <- as.numeric(mathia14[[i]])
}
mathia_cor_success_rate_pairwise_3attempts <- cor(mathia14,
                                                  use = "pairwise.complete.obs")
mathia_cor_success_rate_3attempts <-
  psych::cor.smooth(mathia_cor_success_rate_pairwise_3attempts)
ggm_success_rate_3attempts <- qgraph::qgraph(
      mathia_cor_success_rate_3attempts,
      layout = "spring",
      graph = "glasso",
```

```
    labels = TRUE,
    minimum = 0.1,
    sampleSize = 500
)
```