Measuring the Academic Impact of Course Sequencing using Student Grade Data

Tess Gutenbrunner, Daniel D. Leeds, Spencer Ross, Michael Riad-Zaky, and Gary M. Weiss
Computer and Information Science Department
Fordham University, New York, NY
{tgutenbrunner, dleeds, sross34, mriadzaky, gaweiss}@fordham.edu

ABSTRACT
Undergraduate college students have substantial flexibility in choosing the order in which they take courses, since most courses either have no prerequisites or only a single prerequisite. However, the specific order that courses are taken can have an impact on student performance. This paper describes a general methodology for assessing the impact of course sequencing on student performance, as measured by course grades, and applies this methodology to eight years of undergraduate academic data from Fordham University. The results demonstrate that certain course orderings are associated with improved student grade performance. This study introduces a methodology, new metrics, and a publicly available data-processing tool that can be applied to any student course-grade data set to measure course sequencing effects. The results can be used to inform student decisions, modify course recommendations, and even modify course prerequisites.

Keywords
Data mining, education, course sequencing, student performance

1. INTRODUCTION
Undergraduate university students have substantial flexibility in choosing what courses they take and when they take them. Course sequencing is usually enforced only by a modest set of course prerequisites. This study examines the impact of different course sequences on student learning outcomes, as measured by course grades. The data used in this study includes eight years of undergraduate student grade data from Fordham University. Prior studies on course sequencing have generally been quite limited. Similar research has focused more on course selection, the optimal set of courses for a student to take to maximize performance or time to graduation [4, 5], than on course sequencing. Studies that focused on course sequencing were limited to a single discipline, such as communications [7] and psychology [2]. Our study considers all undergraduate courses within the university, including sequences that span disciplines. Prior studies also only considered how early courses predict performance in later courses, whereas our study does not have this restriction and focuses instead on maximizing overall student performance.

Our study considers the impact of sequencing on pairs of courses. This simplifies the analysis and reduces the risk of finding spurious correlations. The grade performance of students taking each pair of courses in the two possible sequential orderings is measured, with the goal of identifying the ordering that yields the best overall performance (concurrent registrations are excluded from the analysis). Comparing the grade performance for the two sequences required the development of new metrics, which we consider to be one of the contributions of this research. The methodology described in this study, along with the metrics that are introduced, are embodied in a publicly available software analysis tool [6].

Every possible course-pair sequence is considered as long as there are a sufficient number of students to provide reliable results. However, our analysis focuses primarily on course sequences within certain departments and groups of departments. This focus is due to our affiliation with a Computer Science department and the current focus on STEM (Science, Technology, Engineering, and Mathematics) education that is driven by national interests and the needs of industry. We also examine course pairs that include both humanities and STEM courses, because we are interested in the role that a liberal arts education has on STEM education.

There are many factors that can impact instructor performance [8], such as class size, course workload, and time of day of a class [1]. These factors also will impact student performance and hence can interfere with our ability to draw clear conclusions about course sequencing effects. In the present study, we normalize the grade data at the course section level to account for different instructor grading schemes, but do not address the other confounding factors. Our expectation is that the large number of course sections associated with most courses will limit the impact of these factors.

There are several uses for the course sequencing analysis described in this paper. The most obvious is that this information can be used to improve recommendations provided to students concerning beneficial course orderings. When these benefits are substantial enough, official course prerequisites can be modified. Beyond these direct applications of the work, the sequencing results can provide insight into the relationships between courses, and this can be used to inform academic policies. For example, if Course A is not generally considered relevant to Course B, but nonetheless leads to improved student performance in Course B, then one might want to recommend Course A to students who must take Course B.

2. METHODOLOGY
This section describes the data set used, the data preprocessing and transformation that is necessary to convert the data into a form suitable for analysis, and the evaluation metrics that measure the impact of course sequencing.

2.1 Initial Student Course-Grade Data Set
The initial data set describes the grade performance of each undergraduate student in all course sections with at least five students. Each of the 473,527 data set records, which collectively cover 24,969 distinct students, identify a student, a course

(including the course section and semester), the instructor, and the student's grade in the course. Although we aggregate the information to course level, section information is used to normalize student grades. Unfortunately, the initial data set cannot be made publicly available due to strict student privacy laws.

2.2 Data Preprocessing and Transformation

The analysis conducted in this study is based on pairs of courses. From the initial student course-grade data set, we compute and maintain information for each course sequence A→B and B→A, where A and B represent arbitrary courses. For each of these sequences, we maintain a list of all students taking the two courses in the corresponding order, and the grades they receive in each course. The particular section each student enrolls in is also tracked, so that grades can subsequently be normalized at the section level. The transformation of the data from the student course-grade level to the course-pair sequence level, and the generation of our evaluation metrics, are accomplished using our publicly available Python-based tool [6].

In this study, a course pair is analyzed if it meets two conditions. The first condition ensures that the percentage of students taking the sequence in each direction exceeds *MinCSP*, the Minimum Course Sequence Percentage. For this study, *MinCSP* is set to 30%, which ensures that both orderings are taken at least 30% of the time. This excludes abnormal situations where a particular course sequence is rarely taken, such as when a student takes an introductory class in their senior year or retakes a failed course outside of the normal order. The second condition ensures that at least a minimum number of students, *MinCount*, aggregated over all course sections, takes the courses in each order. *MinCount* is utilized to ensure that the sample size is sufficient to generate reliable results. For this study *MinCount* is set to 50 students.

Table 1 specifies how many course pairs remain after these conditions are applied. The conditions are applied sequentially, with *MinCSP* applied before *MinCount*. The values in the rightmost column reflect the number of course pairs actually analyzed. Table 1 displays the number of course pairs for the entire data set, as well as for the five course subsets that are of particular interest to us. Our university has no engineering school, so the STEM courses are offered by the Biology, Chemistry, Computer Science, Mathematics, Natural Sciences, Physics, and Psychology departments. The Humanities courses include all courses from the African and African American Studies, Anthropology, Art History, English, Philosophy, Theology, and Visual Arts departments.

Table 1. Number of course pairs for different course subsets

Data Set	Threshold None	MinCSP=30%	MinCount=50
Full Data Set	81,327	21,461	1,939
Computer Science	850	253	14
Mathematics	392	92	23
Mathematics and CS	1,724	490	51
STEM	12,055	3,000	291
STEM & Humanities	27,303	6,646	684

2.3 Evaluation Metrics

Several metrics are used to analyze the impact of course sequencing on student performance. These metrics are based on lower-level metrics, which are introduced first. Ultimately, we want to see how the mean grades for each course in a course pair are impacted by course order in order to determine the optimal ordering and net benefit in grade performance.

The first step computes the mean grades for each course in a course pair for each of the two orderings. Because instructors vary widely in their leniency when assigning grades, all grades are normalized at the course section level using z-score normalization, as described by Equation 1. In this equation x_i represents the grade of student *i* in the course section, μ represents the mean section grade over x_i, and σ represents the standard deviation of the section grades.

$$Z_i = (x_i - \mu) / \sigma \quad (1)$$

For every course pair <*A*, *B*> we determine the average normalized grade for each course based on each ordering. Specifically, we compute $\mu_A(B \to A)$, $\mu_A(A \to B)$, $\mu_B(A \to B)$, and $\mu_B(B \to A)$, where the subscript of μ denotes the course for which the normalized mean is computed and A → B indicates that course A is taken before course B (and vice versa for B → A). As an example, for the course pair <*Math I*, *English I*>, $\mu_{Math\ I}$(*English I*→*Math I*) represents the mean normalized grade in *Math I* for students who took *Math I* after *English I*.

These normalized means are used to compute the difference in mean normalized grades (DNG). Two DNG values are computed for each course pair <*A*, *B*> since the difference in normalized mean grades is computed for each course. Equations 2 and 3 define these values, where $DNG_{A:B}$ is the difference in mean normalized grade for Course A when Course A is taken after course B rather than before course B, and $DNG_{B:A}$ is the difference in mean normalized grades for Course B when Course B is taken after course A rather than before course A. We compute the difference using the order noted in the equations, because we generally expect a course to perform better when it is taken second and anticipate that most DNG values will be positive.

$$DNG_{A:B} = \mu_A(B \to A) - \mu_A(A \to B) \quad (2)$$

$$DNG_{B:A} = \mu_B(A \to B) - \mu_B(B \to A) \quad (3)$$

The DNG equations measure the benefit of taking two courses in a particular order, but do not reflect the net benefit of one ordering over the other (if both DNG values are positive then the difference between the orderings will be reduced). We therefore compute the order benefit, OB, which is the net difference in DNG values of one ordering over the other. The OB is defined relative to a specific course ordering, as indicated in Equation 4. The OB value will be calculated for both possible orderings, but we will only list the one that is positive, which indicates the optimal course ordering.

$$OB_{A \to B} = DNG_{B:A} - DNG_{A:B} \quad (4)$$

We work through an example using <*Math I*, *English I*>, assuming the following statistics:

$\mu_{Math\ I}(English\ I \to Math\ I)$ = 0.40
$\mu_{Math\ I}(Math\ I \to English\ I)$ = -0.05
$\mu_{English\ I}(Math\ I \to English\ I)$ = 0.40
$\mu_{English\ I}(English\ I \to Math\ I)$ = -0.10

Assuming *Math I* takes on the role of Course A and *English I* Course B, using Equation 2, $DNG_{A:B}$ = 0.40 – (-0.05) = 0.45, and using Equation 3, $DNG_{B:A}$ = 0.40 – (-0.10) = 0.50. Applying Equation 4, we get $OB_{A \to B}$ = 0.50 – 0.45 = 0.05. These results are summarized in the first row of Table 2. The assignment of the two courses to A and B is arbitrary, so we can reverse them, which corresponds to the course ordering in the second row of Table 2. Then, using Equation 2 and Equation 3, we get $DNG_{A:B}$ = 0.50 and $DNG_{B:A}$= 0.45, which yields an OB value of 0.45 – 50 = -0.05. The values of $DNG_{A:B}$ and $DNG_{B:A}$ in Table 2 are flipped when we reverse the roles of A and B (compare rows 1 and 2). This is logically and mathematically required given the definition of the DNG metric, so the OB value of one ordering must equal the negative of the other. The results in Table 2 show that taking *Math I* and then *English I* yields an overall improvement in normalized grades of 0.05, whereas taking the courses in the reverse order yields a net deterioration of 0.05.

Table 2. Example of a course pairing

Course A	Course B	$DNG_{A:B}$	$DNG_{B:A}$	$OB_{A \rightarrow B}$
Math I	English I	0.45	0.50	0.05
English I	Math I	0.50	0.45	-0.05

3. RESULTS

This section provides selected results from our analysis, with a focus on the difference in normalized grades for different course sequences. Order benefit is our primary metric, as it summarizes the net benefit of a particular course sequence over the alternative, but DNG is also informative since it specifies the amount of benefit in taking one course before the other. For example, it is possible for two competing sequences to have identical positive DNGs, leading to a zero order benefit. Top order benefit results are presented for course sequences restricted to: Computer Science, Math, Math and Computer Science, STEM, STEM and Humanities, and "All Courses" across all disciplines. We posit explanations for some of the results based on our knowledge of the domain.

The top three order benefit values for computer science courses are displayed in Table 3. Note that while the sequence *Computer Algorithms* → *Data Mining* has the highest OB value, based on the $DNG_{B:A}$ values, taking *Data Communications and Networks* after *Data Mining* yields a slightly greater improvement than taking *Data Mining* after *Computer Algorithms*. The key difference is that taking each of those pairs of courses in the opposite order (i.e., $DNG_{A:B}$) yields very different results. The two negative $DNG_{A:B}$ values in Table 3 indicate that the corresponding courses yield worse results when they are taken second. Specifically, students in *Computer Algorithms* perform worse when they take it second. We generally would not expect this to occur. This result may stem from weaker students who delay taking *Computer Algorithms*.

Table 3. Computer Science courses with largest order benefit

Course A	Course B	$DNG_{A:B}$	$DNG_{B:A}$	OB
Computer Alg.	Data Mining	-0.110	0.233	0.343
Data Structures	Computer Organization	-0.073	0.103	0.176
Data Mining	Data Comm. & Netwks.	0.101	0.235	0.134

A plausible explanation for the first entry in Table 3 is that *Data Mining* utilizes some knowledge of *Computer Algorithms* and hence taking *Data Mining* second is beneficial. While the same reasoning could be applied to the reverse ordering, the negative DNG indicates no benefit for that ordering, possibly because *Data Mining* does not teach the basics of computer algorithms. With respect to the entry in the second row of Table 3, the benefit of foundational mathematics and algorithmic knowledge provided by *Data Structures* is apparent in the somewhat more application-oriented *Computer Organization* course.

Table 3 shows negative $DNG_{A:B}$ values are smaller in magnitude than positive $DNG_{B:A}$ values — a finding replicated in subsequent tables. The presence of negative $DNG_{A:B}$ values may be an artifact of our focus on course pairs with the highest overall order benefit, because order benefit is maximized when $DNG_{A:B}$ is negative.

Table 4 shows the results for three sequences of mathematics courses. The third entry is the easiest to explain. *Business Finite Math* and *Finite Math* cover similar material, but the former covers more basic material. Students are not generally expected to take both courses, but if they do, they most likely will take the more basic one first. *Discrete Math* provides a background in formal proofs, which appears to benefit from advanced mathematical experience (*Multivariable Calculus I*) and to provide benefit to advanced study of calculus (*Multivariable Calculus II*).

Table 4. Mathematics courses with largest order benefit

Course A	Course B	$DNG_{A:B}$	$DNG_{B:A}$	OB
Discrete Math	Multivar. Calc II	-0.056	0.252	0.308
Multivar. Calc. I	Discrete Math	-0.041	0.249	0.290
Business Finite Math	Finite Math	-0.024	0.145	0.169

Most computer science programs require several mathematics courses, but the specific impact of the math courses on computer science courses is not well understood. Table 5 explores the relation between the two departments, restricting the sequences to include one math course and computer science course. One of the more notable results is the entry in the first row. Both courses teach finite mathematics, but *Structures of Computer Science* is offered by the Computer Science department and is intended for non-majors, while *Finite Math* is offered by the Mathematics department. *Structures of Computer Science* also devotes several weeks to cover simple programming assignments, thereby further reducing the time spent on the mathematics content. For these reasons, it is reasonable to conclude that the sequence with the high OB value corresponds to taking the more basic course first. It is also noteworthy that *Calculus I* has a very positive impact on taking programming courses (*Computer Science I* and its lab) and *Structures of Computer Science*. Thus it appears that increased mathematical sophistication does have a positive impact on computer science and computer programming. This is especially interesting because the mathematical material in *Calculus I* has only a tangential relationship with computer science. Most computer science programs require calculus, and our empirical data justifies this requirement.

Table 5. Math and CS courses with largest order benefit

Course A	Course B	$DNG_{A:B}$	$DNG_{B:A}$	OB
Structures of CS	Finite Math	-0.002	0.429	0.431
Calculus I	CS I	-0.035	0.338	0.373
Calculus I	CS I Lab	-0.012	0.252	0.264
Calculus I	Structures of CS	-0.010	0.213	0.223

Table 6 displays the remaining results for the three groupings of sequences: STEM courses, mixed STEM and humanities courses, and all courses without any restrictions. The first entry under the STEM category shows a benefit in taking *Applied Calculus I* after *General Chemistry I*. This ordering is typical for students on the Pre-Health track who wish to go to medical school, which may explain the high order benefit, since these students are generally motivated to achieve high grades. Furthermore, under the STEM category we find a benefit for *Learning (Psychology)* followed by *Multicultural Psychology*. The first psychology course in this sequence is a 2000 level course while the second is a 3000 level course, indicating yet again that there is a benefit from taking a more advanced course in the same discipline second.

Looking at the STEM & Humanities courses, students who took *Organic Chemistry I→ Intro. to Cultural Anthropology* did significantly better in both classes, as demonstrated by the magnitudes of the DNG values (the negative $DNG_{A:B}$ indicates *Organic Chemistry I* does worse when taken second and hence performs better when taken first). The same pattern is replicated with an even higher OB when considering the *Organic Chemistry Lab*. Pre-Health students tend to take *Organic Chemistry* very early in their college career and may dominate that particular course ordering.

The first row under the "All Courses" category displays the sequence *Spanish Language & Literature → Christian Hymns* with a very high order benefit. Students performed best in each of the two courses when taking them in the specified sequence. This may be due to the fact that Spanish literature is heavily influenced by Christianity, and therefore provides important background for students who plan to take *Christian Hymns*. Explanations for the other entries may require consultation with faculty from the associated departments.

Table 6. STEM, STEM/Humanities, All courses with large OB

Course A	Course B	$DNG_{A:B}$	$DNG_{B:A}$	OB
STEM Courses				
General Chem. I	Applied Calculus I	-0.17	0.400	0.570
Intro. Astronomy	Abnormal Psych.	-0.187	0.309	0.496
Learning (Psych.)	Multicultural Psych.	-0.021	0.419	0.440
Intro. Bio. I	Structures of CS	-0.152	0.283	0.435
Structures of CS	Finite Math	-0.002	0.429	0.431
Gen. Chem. Lab I	Structures of CS	-0.102	0.325	0.427
Calculus I	CS I	-0.035	0.338	0.373
Intro. Bio Lab I	Structures of CS	-0.069	0.297	0.366
Physics II Lab	Human Physiol. Lab	-0.019	0.287	0.306
STEM & Humanities Courses				
Org. Chem. Lab I	Intro. Cultural Anthr.	-0.520	0.606	1.126
Organic Chem. I	Intro. Cultural Anthr.	-0.330	0.554	0.884
Forensic Science	Philosophical Ethics	-0.310	0.474	0.784
Texts & Contexts	Discrete Math	-0.234	0.372	0.606
All Courses				
Spanish Lang. & Lit.	Christian Hymns	-0.436	0.714	1.150
Medieval History	Intro. Media Industry	-0.178	0.550	0.728
Composition II	Intro. Archaeology	-0.218	0.494	0.712
Sociology Focus	Faith & Crit. Reason	-0.134	0.565	0.699
Calculus II	Intro Sociology	-0.111	0.488	0.599
American History	Personality (Psych)	-0.095	0.487	0.582

4. CONCLUSION

The research described in this study introduced a methodology and set of metrics for assessing the impact of course sequencing on student performance. The analysis of our results focuses on several disciplines, such as Computer Science and Mathematics, as well as higher level groupings, such as STEM courses. Many of the results demonstrate that there is a substantial benefit with a particular sequencing of courses, such as taking *Finite Math* after *Structures of Computer Science* or taking *Computer Science I* after *Calculus I*. Our methodology and metrics are implemented in our Python-based software tool [6], which can be used by other researchers.

The course sequencing results in this paper can be used to assist with course recommendations and can be used to inform, and even modify, course prerequisites. For example, our results show a larger than expected benefit of taking calculus before a programming course; additional analysis and data will be needed to see if this extends to a broader set of mathematics courses, but if it does, then new prerequisites perhaps should be added. The results in this study also provide insight into the inter-relationships between courses and disciplines.

Many of our observed results can be explained based on our knowledge about college education and domain knowledge of specific disciplines. However, in some cases explanations are not readily available. Our search for explanations of why one sequence may outperform another can also benefit from additional domain knowledge, as our knowledge is mainly limited to computer science. Course syllabi could also prove to be useful. It would also be very interesting to apply our methodology to data from different universities, and we hope to do this in the future. It would be informative to see if the course sequencing patterns present in our university hold elsewhere. Although our university is relatively large, in many cases the number of students taking some pairs of courses was relatively small, and this informed our relatively low *MinCount* threshold of 50. With more data, we could increase this threshold, which would diminish the impact of factors like instructor effectiveness.

Our methodology normalizes for some external factors, such as different instructor grading schemes, but does not account for all factors that can impact student performance. In particular, we suspect that some course sequencing results are due to certain populations of students (e.g., Pre-Health students) taking courses in one particular order over another. In future work we do plan to consider some of these factors and modify our evaluation to isolate their impact. In cases where that is not feasible, we will at least provide summary statistics to assess the influence of these factors. For example, since we suspect that academically stronger students sometimes take courses in a different sequence than weaker or less motivated students, we can compare the overall GPAs of students taking the courses in each course ordering and note when they exhibit a statistically significant difference. Alternatively, we can normalize for overall student GPA, something that we are currently doing in a study on instructor effectiveness.

One final area that we plan to pursue is better evaluation of our results. One way to do this is to utilize statistical significance testing. Given the number of potential patterns we can find with the large number of pairs of courses, we may need to set our p-value quite low. We may be able to improve this situation by limiting our course interactions to courses within a single department or between related fields (e.g., Biology and Chemistry). We can also validate our results by partitioning the data into a training and test set and subsequently verifying if the patterns found in the training data hold for the test data. In this regard, the differences in student performance can be viewed as predictions, so the standard training and test set evaluation methodology applies.

The data utilized in this study is itself a valuable resource. Our research group has analyzed this data in a variety of ways to provide additional insights. Two studies have used this data to group/cluster courses and analyze the interrelationships between courses. One of these studies uses course co-enrollments to form the clusters and to identify hub courses [9], while the other uses the correlation between students grades as a similarity metric to cluster the courses [3]. Both of these studies used their respective notions of similarity to form networks of courses, and then analyzed these with existing network analysis techniques.

5. REFERENCES

[1] Anderson, C. J., Poulsen, S., and West, M. 2020. The relationship between course scheduling and student performance. In *Proceedings of the 4th Educational Data Mining in Computer Science Education Workshop* (collocated with EDM2020), July 10, 2020.

[2] Betancur, L., Rottman, B. M., Votruba-Drzal, E., and Schunn, C. 2019. Analytical assessment of course sequencing: The case of methodological courses in psychology. *Journal of Educational Psychology*, 111(1), 91.

[3] Leeds, D. D., Zhang, T., and Weiss, G. M. 2021. Mining course groupings using academic performance. In *Proceedings of the 14th International Conference on Educational Data Mining*.

[4] Morsy. S., and Karypis, G. 2019. Will this course increase or decrease your GPA? Towards grade-aware course recommendation. *Journal of Educational Data Mining*, 11(2):20–46, 2019.

[5] Parks, M. R., Faw, M., & Goldsmith, D. 2011. Undergraduate instruction in empirical research methods in communication: Assessment and recommendations. *Communication Education*, 60, 406-421. DOI= 10.1080/03634523.2011.562909

[6] Riad-Zaky, M., Weiss, G.M., and Leeds, D.D. Course Grade Analytics with Networks (CGAN) [computer software], *Available: http://www.cis.fordham.edu/edmlab/software*

[7] Richards, A. S. 2012. Course sequencing in the communication curriculum: A case study. *Communication Education*, 61(4), 395-427. DOI= 10.1080/03634523.2012.713500

[8] Wachtel, H. K. 1998. Student Evaluation of College Teaching Effectiveness: a brief review, *Assessment & Evaluation in Higher Education*, 23:2, 191-212, DOI= 10.1080/0260293980230207

[9] Weiss, G.M., Nguyen, N., Dominguez, K., and Leeds, D.D. 2021. Identifying hubs in undergraduate course networks based on scaled co-enrollments. In *Proceedings of the 14th International Conference on Educational Data Mining*.