California Community Colleges

AB705 Success Rates Estimates
Technical Paper

Estimating Success Rates for Students Placed Directly into Transfer-Level English and Math Courses

MMAP Team

September 2018

rpgroup.org

EDUCATIONAL RESULTS PARTNERSHIP

theRPgroup
Research • Planning • Professional Development for California Community Colleges

Table of Contents

Executive Summary

California Assembly Bill (AB) 705 authored by Irwin and passed on October 13, 2017, requires colleges to "maximize the probability that a student will enter and complete transfer-level coursework in English and math within a one-year timeframe" and use high school background data in placement processes. To implement this new law and develop guidelines based on the best available evidence, the California Community College Chancellor's Office (CCCCO) created the AB 705 Implementation Committee (Committee). One key request from the Committee was to focus on students historically placed into below transfer-level courses and compare transfer-level success rate estimates if they were placed directly into transfer-level coursework to the estimates for those who started one level below transfer-level using data from the Multiple Measures Assessment Project (MMAP). Compared to students who were placed directly into transfer-level courses, students with similar high school backgrounds but who had not historically been placed in transfer-level coursework may have had lower placement test scores or high school performance, so theoretically might not perform as well if placed there directly.

A series of regressions using high school grade point average (HSGPA) and ACCUPLACER scores were used to adjust direct transfer-level placement success rates for three gatekeeper classes: transfer-level English, statistics, and pre-calculus. These estimated success rates were then compared to estimated "throughput" rates (the percentage of students completing transfer-level English or math in a given time frame) of students placed one level below to determine if such remediation would result in higher transfer-level completion or throughput than direct placement into transfer-level coursework. The regression-adjusted success rates were indeed lower than the original success rates of students who had been placed directly into a transfer-level course in the MMAP decision rules data. However, for all HSGPA performance levels in all three gatekeeper courses, the adjusted success rates for students placed directly into transfer-level courses exceeded adjusted throughput rates for students placed one level below transfer. This result suggests that **even without any additional supports or course redesigns, the lowest performing high school students would have been more likely to complete transfer-level English, statistics, or pre-calculus if placed directly into these courses as compared to taking below transfer-level remediation**. We were unable to identify any group of students who complete the transfer-level English, statistics, or pre-calculus course at a lower rate when placed directly there as opposed to being first placed in courses that are below transfer-level. It is recommended that each college conduct its own analysis to compare throughput rates below transfer-level to success rates at transfer-level at each level of high school achievement. These analyses should also be disaggregated by gender and ethnicity, both with and without specialized support, such as co-requisites, to ensure that local data align with the statewide findings. Further, colleges are encouraged to evaluate and assess their placement processes, curricular design, concurrent supports, and non-curricular supports, as well as determine and address disproportionate outcomes for historically underrepresented populations.

These findings were used to inform guidance memos from the AB 705 Implementation Committee. This document provides details on how these adjustments were made, including the analytical code to transparently document methods and support local replication.

Introduction

The Multiple Measures Assessment Project (MMAP)[1] is a collaborative effort led by the Research and Planning Group for California Community Colleges (RP Group) and the Educational Results Partnerships' (ERP) Cal-PASS Plus system, with input and support from the California Community College Chancellor's Office (CCCCO) and the Academic Senate for California Community Colleges (ASCCC). MMAP seeks to develop, pilot, and assess implementation of enhanced multiple measures, including the use of high school performance and non-cognitive variables, in student placement and advising. This project was part of the California Community College's Common Assessment Initiative (CAI) and now is supporting the implementation of AB 705,[2] which requires "that a community college district or college maximize the probability that a student will enter and complete transfer-level coursework in English and math within a one-year timeframe and use, in the placement of students into English and math courses, one or more of the following: high school coursework, high school grades, and high school grade point average."

Early phases of the Multiple Measures Assessment Project placement recommendations were driven by the goal of identifying students who were highly likely to succeed at any given level of a communication or computation sequence (e.g., English, English as a Second Language [ESL], reading, and math)[3]. The results suggested that many students had been placed at too low of a level, with students of color and female students placed disproportionately lower, especially in math.

The passage of California Assembly Bill (AB) 705 required the field to respond to a different question: how can we maximize the likelihood that students successfully complete gateway transfer-level English and math in one year and ESL in three years?

Estimating and Maximizing Throughput

In this context, throughput rate is defined as the percentage of students who complete transfer-level English or math with a grade of C or better within one year (two semesters or three quarters). The throughput rate for students placed directly into transfer-level courses consists of the success rate (grade of C or better) in the first transfer-level course enrolled.

For students whose first attempt was below transfer-level, the throughput rate is defined as the percentage of these students who successfully complete the first attempted transfer-level course in the sequence within one year. Note this definition does not require the below transfer level course to have been successfully completed in the data set as there are several ways by which a student could become eligible to progress into transfer level that would not appear in the database. These include but are not limited to completing coursework at a college not in the data system or a challenge process. In addition, this definition includes only the first transfer level attempt and does not include subsequent attempts due to the structure of the MMAP data file. Local replications with access to more detailed student records could include tracking of equivalencies brought in from other colleges and repeated attempts.

1 http://rpgroup.org/All-Projects/ArticleView/articleId/118/Multiple-Measures-Assessment-Project-MMAP

2 https://assessment.cccco.edu/

3 http://rpgroup.org/Portals/0/Documents/Projects/MultipleMeasures/DecisionRulesandAnalysisCode/Statewide-Decision-Rules-5_18_16_1.pdf

Rather than identifying students who are highly *likely* to succeed, AB 705 changes the task to identifying students highly *unlikely* to succeed if directly placed into transfer-level courses and maximize their relative likelihood of completing that course if they start at transfer-level or below transfer-level. The empirical challenge is to identify a group of students who are more likely to complete the transfer-level gateway course if they begin in the basic skills sequence rather than directly in transfer-level courses.

Examining the latest MMAP decision trees[4] suggests that the group of students most likely to benefit from traditional basic skills sequences would be those with the lowest likelihoods of success in transfer-level coursework. These students also have lower high school grade point averages (HSGPA). In the MMAP rule sets designed before the passage of AB 705, it was recommended that these students enroll in classes one or more levels below transfer to increase the likelihood of success in the target class and to maintain success rates similar to the previous, pre-MMAP success rates. However, with AB 705's focus on throughput rates, there is a need to compare the throughput rate from these classes that are one or more levels below transfer-level to throughput rates from direct transfer-level placement. In particular, for lower high school GPA students, the AB 705 Implementation Committee inquired would their completion rates of transfer-level courses be maximized by direct placement into transfer-level or placement into a below transfer-level course?

Using a transfer-level statistics course as an example, the MMAP decision rules indicated that students with a high school GPA of 2.3 or less (unweighted HSGPA up through 11th grade) previously had success rates of 40% when placed directly into transfer-level statistics without additional supports. The estimated throughput rate for these same students beginning one level below (e.g., intermediate algebra) is approximately 8%, which suggests that even though the success rates are low with direct placement (40%), the throughput rate is still higher than that realized by traditional remediation (8%).

To create the one level below transfer-level throughput estimates for both math and English, the MMAP team examined students who began one level below transfer and tracked their course-taking for two primary semesters or three primary quarters from that initial one level below course (including any intervening intersessions). This data set allowed the team to calculate the proportion of students who successfully completed a transfer-level course within one year of their initial attempt. Moreover, throughput rates were calculated for distinct sets of students who were described in the terminal nodes of the transfer-level MMAP decision trees, with a particular focus on the students with the lowest probability of transfer-level success. Students who were the lowest performing in high school received particular focus and had throughput rates calculated as a subgroup since they theoretically would be most likely to benefit from below transfer-level remediation.

For math, the throughput rates for statistics and pre-calculus were adjusted upward to account for intent to earn a certificate/degree or transfer to a four-year college along either a statistics, liberal arts, and math (SLAM) pathway or a science, technology, engineering, or math (STEM) pathway. Based on national research on major intent,[5] the MMAP team assumed that 25% of students had a STEM intent and 75% had a SLAM intent. See Appendix A for R code used to categorize courses and pathways.

---

4 http://rpgroup.org/Portals/0/Documents/Projects/MultipleMeasures/DecisionRulesandAnalysisCode/English-Decision-Trees-1_11_2016.pdf
http://rpgroup.org/Portals/0/Documents/Projects/MultipleMeasures/DecisionRulesandAnalysisCode/Math-Decision-Trees-4_3_16.pdf

5 https://www.nsf.gov/statistics/2016/nsb20161/#/#/report/chapter-1/transition-to-higher-education/preparation-for-college

Methods

The MMAP research is based on retrospective data files—data collected from students who have already been placed, enrolled, and attempted courses in the community college system. The 40% success rate in the statistics example used earlier is based upon the rates of success of students with low HSGPAs who were also eligible to enroll into transfer-level statistics. There are several ways in which a low HSGPA student could have received a transfer-level placement, including, but not limited to:

1. Scoring sufficiently high on a placement test (through skill or through chance);
2. The college's acceptance of a placement from another college with different cut scores;
3. The student having locally accepted alternative assessments, such as scores on the Early Assessment Project (EAP), Scholastic Aptitude Test (SAT), American College Test (ACT), or Advanced Placement (AP) exams;
4. Transferring in coursework from other institutions not present in the data set;
5. Successfully challenging the prerequisite through the local college process; or
6. An error in the intake process.

Given the data caveats above, there is the possibility that selection bias may occur. Selection bias takes place when the selection of the students to be analyzed is undertaken in such a way that the sample is not representative of the population intended to be analyzed. For example, lower-performing high school students who are also placed into transfer-level math courses may not be representative of all lower-performing high school students. Thus, the success rates those students achieved may not be representative of the success rates of lower-performing high school students who did not also place into a transfer-level math course. The 40% success rate in statistics courses noted previously could then be an overestimation of other low HSGPA students' likelihood of success in statistics if placed there directly, unless one adjusts for differences related to students' performance that are associated with their selection into various placement levels.

To address possible selection bias, the MMAP research team conducted regression analyses using the same data set that informed the development of the MMAP decision rule placement recommendations. Data were analyzed for low HSGPA students enrolled in three gateway transfer-level courses (transfer-level English, statistics, and pre-calculus).[6]

While students can be placed according to a variety of means, scores on standardized tests have been the most common placement tool and are heavily weighted in overall placement practices in California Community Colleges (Regional Education Laboratory [REL] WestEd, 2011). The MMAP research team obtained ACCUPLACER scores for a subset of English and math students. ACCUPLACER was an approved standardized test during the timeframe of the data and was widely used for placement. It should be noted that due to the low predictive validity of ACCUPLACER relative to high school performance measures and the limited number of students for whom the scores were available, the first phase of MMAP did not use ACCUPLACER scores in the first draft of rule sets; the second-phase MMAP rule sets did not utilize ACCUPLACER data at all as the goal was to develop multiple measures that could operate independently of standardized tests. However, these scores are important for understanding the magnitude of possible selection biases because of the central role they played in placement.

6 ESL placement is more complex and will be addressed in subsequent documents.

The overall approach to estimating direct placement success rates for all low HSGPA students in the target course was as follows:

1. Fit a regression model that predicts success in target course based on high school GPA and ACCUPLACER test scores.
2. Calculate mean high school GPA and test scores for lowest-node students in each level (or each type, for transfer-level math) of first attempted course.
3. Use regression model from step 1 to predict success in the target course for each level of first attempted course using means in step 2.
4. Rescale regression-predicted success rates against the lowest-node predicted success rates to create comparability between decision-tree and regression-based predictions.
5. Calculate an overall success rate estimate by weighting estimates from each level (or type) of first attempted course weighted by number of students beginning at each level.
6. Use standard error of prediction from the regression model at each level to create lower- and upper-error bounds for estimates also weighted as in step 5.

In the testing phase, both logistic and linear regressions were used, as the outcome was a binary success indicator ("1" indicated earning a grade of C or better, and "0" indicated earning a non-satisfactory grade such as NP, D, F, or W). While logistic regression is typically used for dichotomous outcomes, results from the ordinary least squares (OLS) model are more accessible to a broader audience and yielded findings comparable to logistic regression in terms of overall model significance and relative predictive strength of input variables (Hellevik, 2007; Cohen and Cohen, 1983; Pindyck and Rubinfeld, 1981). The ease of interpretation resulted in the selection of linear regression to create estimates.[7]

7 The actual code used along with regression coefficients, mean values, and excel formulas for estimates and weighting are provided in Appendix B.

Caveats

A challenge arose in creating these estimates that necessitated slightly different approaches for English versus math courses. While English has only one gateway transfer-level course, the progression from remedial math into transfer-level math is split into different math pathways and multiple possible transfer-level math courses. For statistics and pre-calculus, all cases were combined, regardless of HSGPA, due to the relatively small number of cases with test scores for each of the different transfer-level math options. The data set contained scores from three levels of ACCUPLACER math: arithmetic, elementary algebra, and college algebra. The college algebra test scores were used to both maximize the number of cases available and utilize the test level most appropriate to transfer-level placement. The college algebra test was a significant predictor of success in both college statistics and pre-calculus, although the relative effect size was much smaller than that of high school GPA.

While this analysis relied upon ACCUPLACER scores, there are other standardized placement tests, which may have a better, similar, or worse predictive validity than ACCUPLACER. There may also be associations between the type of test used and characteristics of the college and students that theoretically could strengthen or weaken the regression models.

Additionally, the MMAP data set consists of students who were taught under the previous California K-12 standards, while future students increasingly will have experienced the new Common Core K-12 standards. The data set was also limited to high schools that provide data to CalPASS Plus. Although the data file used was a large and generally representative data set for statewide purposes, the file did not contain all students, which creates limitations when examining localized areas with low CalPASS Plus participation. A centralized statewide system containing all California K-12 and postsecondary data would be greatly beneficial to future research and the implementation of AB 705 requirements.

Results

The regressions for each target course produced a series of estimates for the success rates of students placed directly into the transfer-level course, with error bounds for "lowest-node" students (see Appendices B through E for technical details and estimates for other nodes). The regression R2 was 0.10 for transfer-level English, 0.10 for statistics, and 0.09 for pre-calculus. This suggests acceptably strong models in a social science context.

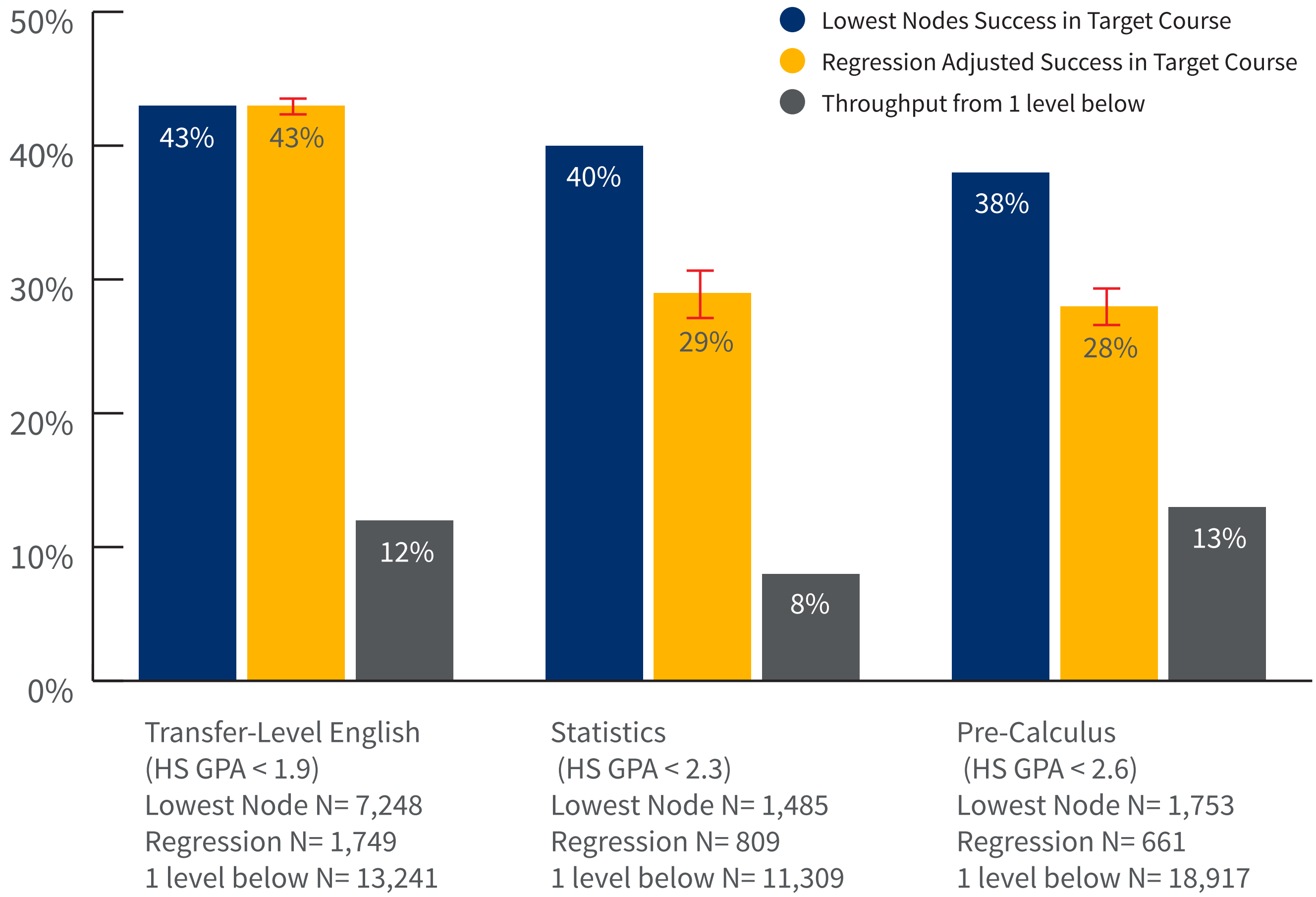
Figure 1 on the following page displays the lowest-node success in each target gateway course from the MMAP decision tree analyses, the regression-adjusted success estimates in each target course, and the estimated throughput from one level below transfer-level for each target course. This figure demonstrates several key points:

- For transfer-level English, the lowest node success was 43%, while the regression-adjusted success rate point estimate was 42.6% (rounding to 43%), with a standard error of 0.9% and an estimated throughput from one level below of 12%. The regression adjustment for transfer-level English was very small and did not result in a practical difference from the original decision tree estimate. Both of these transfer-level success estimates for transfer-level English were over three times greater than the throughput estimate from one level below.

- In statistics, the lowest node success was 40%, while the regression-adjusted success rate point estimate was 29%, with a standard error of 3.7%, and an estimated throughput from one level below of 8%. The regression-adjusted direct transfer-level placement success rate for statistics was over three times greater than the one level below transfer-level throughput estimate.

- With pre-calculus, the lowest node success was 38%, while the regression-adjusted success rate point estimate was 28%, with a standard error of 3.1%, and an estimated throughput from one level below of 13%. The regression-adjusted direct transfer-level placement success rate for pre-calculus was over twice the value of the one level below transfer-level throughput estimate.

FIGURE 1. TRANSFER LEVEL SUCCESS RATES FOR LOWEST NODE HIGH SCHOOL GPA STUDENTS FROM DECISION TREE ANALYSES AND WITH REGRESSION ADJUSTMENTS COMPARED TO ESTIMATED THROUGHPUT RATE FROM ONE LEVEL BELOW TRANSFER. RED BARS REPRESENT ± 1 STANDARD ERROR.



For all three gateway transfer-level courses, the adjusted success rate point estimates were lower than the success rates from the lowest-node decision tree analyses, but in all cases were well above the estimated throughput from one level below transfer-level. **This result suggests that even without any additional supports or course redesigns, the lowest performing high school students would have been more likely to complete transfer-level English, statistics, or pre-calculus if placed directly into these courses as compared to taking below transfer-level remediation**. Appendix E provides estimates for other high school performance levels.

Conclusion

This analysis did not find evidence that students would have higher throughput rates by being placed into basic skills courses based on their high school performance. Thus, within the timeframe of data availability, and given the curricular design and support structures that existed systemwide at this time, we were unable to identify any group of students who complete the transfer-level English, statistics, or pre-calculus course at a lower rate when placed directly there as opposed to being first placed in courses that are below transfer-level.

It is recommended that each college conduct its own analysis to compare throughput rates below transfer-level to success rates at transfer-level at each level of high school achievement. These analyses should also be disaggregated by gender and ethnicity, both with and without specialized support, such as co-requisites, to ensure that local data align with the statewide findings. Further, colleges are encouraged to evaluate and assess their placement processes, curricular design, concurrent supports, and non-curricular supports, as well as determine and address disproportionate outcomes for historically underrepresented populations.

References

Cohen, J., & Cohen, P. (1983). Applied multiple regression/correlation analysis for the behavioral science. London: Lawrence Erlbaum.

Hellevik, O. (2007). Linear versus logistic regression when the dependent variable is a dichotomy. Quality and Quantity. 43, 59–74.

MMAP Research Team. (2017). Multiple Measures High School Variables Model Summary – Phase II. San Rafael, CA: The Research and Planning Group for California Community Colleges. Retrieved April 28, 2018 from http://rpgroup.org/Portals/0/Documents/Projects/MultipleMeasures/DecisionRulesandAnalysisCode/Statewide-Decision-Rules-5_18_16_1.pdf

Pindyck, R., & Rubinfeld, D. (1981). Econometric models and economic forecasts. New York: McGraw-Hill.

REL West at WestEd. (2011). 1.2.110 Types of Multiple Measures Used in California Community College Mathematics, English, and English as a Second Language Course Placement: Summary Report of Survey Results. San Francisco, CA: WestEd.

Appendix A: R code for estimating one-year throughput rates for Enlgish and math

To access both an R tutorial and the R code used in the MMAP analyses, visit:

- MMAP code: http://bit.ly/MMAP_code
- R Tutorial: http://bit.ly/R_Tutorial

```
#statistics
math1111$CC_STATISTICS_00 <- 0
math1111$CC_STATISTICS_00[grepl("sta",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE] <- 1
math1111$CC_STATISTICS_00[grepl("MATH 123",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE] <- 1

table(math1111$CC_STATISTICS_00)

#business math
math1111$CC_BUS_MATH_00 <- 0
math1111$CC_BUS_MATH_00[grepl("bus",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE] <- 1
#business stats coded as statistics
math1111$CC_BUS_MATH_00[grepl("stat",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE] <- 0
#business finite math coded as finite
math1111$CC_BUS_MATH_00[grepl("finite",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE] <- 0

table(math1111$CC_BUS_MATH_00)

#Liberal Arts math
math1111$CC_LA_MATH_00 <- 0
math1111$CC_LA_MATH_00[grepl("MATH FOR",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE] <- 1
math1111$CC_LA_MATH_00[grepl("MATH/LIB",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE] <- 1
math1111$CC_LA_MATH_00[grepl("MATH/ELEM",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE] <- 1
math1111$CC_LA_MATH_00[grepl("MATH:ELEM",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE] <- 1
math1111$CC_LA_MATH_00[grepl("MATH TOPICS",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE] <- 1
math1111$CC_LA_MATH_00[grepl("PRINCIPLES OF MATHEMATICS",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE] <- 1
math1111$CC_LA_MATH_00[grepl("LIBERAL",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE] <- 1
math1111$CC_LA_MATH_00[grepl("SURVEY",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE] <- 1
math1111$CC_LA_MATH_00[grepl("CONCEPTS",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE] <- 1
math1111$CC_LA_MATH_00[grepl("IDEAS",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE] <- 1
math1111$CC_LA_MATH_00[grepl("TEACH",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE] <- 1
math1111$CC_LA_MATH_00[grepl("GEN",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE] <- 1
math1111$CC_LA_MATH_00[grepl("DISCOVERY",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE] <- 1
math1111$CC_LA_MATH_00[grepl("REASONING",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE] <- 1
math1111$CC_LA_MATH_00[grepl("SYMBOLIC",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE] <- 1
math1111$CC_LA_MATH_00[grepl("STRUCTURE",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE] <- 1
math1111$CC_LA_MATH_00[grepl("NATURE",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE] <- 1
math1111$CC_LA_MATH_00[grepl("TCHRS",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE] <- 1
math1111$CC_LA_MATH_00[grepl("EXPLOR",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE] <- 1
math1111$CC_LA_MATH_00[grepl("IN ELEM MATH",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE] <- 1
math1111$CC_LA_MATH_00[grepl("PATTERNS",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE] <- 1
math1111$CC_LA_MATH_00[grepl("CONTEMPORARY",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE] <- 1
math1111$CC_LA_MATH_00[grepl("Intro-Discrete Structures",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE] <- 0
math1111$CC_LA_MATH_00[grepl("Math for Business",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE] <- 0
math1111$CC_LA_MATH_00[grepl("calculus",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE] <- 0

table(math1111$CC_LA_MATH_00)

#college algebra
math1111$CC_COLL_ALG_00 <- 0
math1111$CC_COLL_ALG_00[grepl("coll",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE] <- 1
math1111$CC_COLL_ALG_00[grepl("advanced alg",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE] <- 1
math1111$CC_COLL_ALG_00[grepl("math 131",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE] <- 1
math1111$CC_COLL_ALG_00[math1111$CC_LA_MATH_00==1] <- 0
```

```
table(math1111$CC_COLL_ALG_00)

#Precalc
math1111$CC_PRE_CALC_00 <- 0
math1111$CC_PRE_CALC_00[grepl("pre",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE] <- 1
math1111$CC_PRE_CALC_00[grepl("COORDINATE GEOMETRY",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE] <- 1
math1111$CC_PRE_CALC_00[grepl("MATH 141",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE] <- 1
math1111$CC_PRE_CALC_00[grepl("MATH041",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE] <- 1

table(math1111$CC_PRE_CALC_00)

#calc II
math1111$CC_CALC_II_00 <- 0
math1111$CC_CALC_II_00[grepl("II",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE & grepl("calc",math1111$CC_00_
COURSE_TITLE,ignore.case=TRUE)==TRUE] <- 1
math1111$CC_CALC_II_00[grepl("2",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE & grepl("calc",math1111$CC_00_
COURSE_TITLE,ignore.case=TRUE)==TRUE] <- 1
math1111$CC_CALC_II_00[grepl("3",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE & grepl("calc",math1111$CC_00_
COURSE_TITLE,ignore.case=TRUE)==TRUE] <- 1
math1111$CC_CALC_II_00[grepl("MATH ANALYSIS III",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE] <- 1
math1111$CC_CALC_II_00[grepl("MATHEMATICAL ANALYSIS III",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE] <- 1
math1111$CC_CALC_II_00[grepl("MATH ANALYSIS II",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE] <- 1
math1111$CC_CALC_II_00[grepl("MATHEMATICAL ANALYSIS II",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE] <- 1
math1111$CC_CALC_II_00[grepl("MATHEMATICAL ANALYSIS 2",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE] <- 1
math1111$CC_CALC_II_00[grepl("MATHEMATICAL  ANALYSIS 2",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE] <- 1
math1111$CC_CALC_II_00[grepl("ANALYTIC",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE &
grepl("II",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE] <- 1
math1111$CC_CALC_II_00[grepl("ANALYTIC",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE &
grepl("2",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE] <- 1
math1111$CC_CALC_II_00[grepl("ANALYTIC",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE &
grepl("3",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE] <- 1
math1111$CC_CALC_II_00[grepl("math182",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE] <- 1

table(math1111$CC_CALC_II_00)

#calculus I
math1111$CC_CALC_I_00 <- 0
math1111$CC_CALC_I_00[grepl("calc",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE] <- 1
math1111$CC_CALC_I_00[grepl("MATH ANALYSIS",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE] <- 1
math1111$CC_CALC_I_00[grepl("MATHEMATICAL ANALYSIS",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE] <- 1
math1111$CC_CALC_I_00[grepl("ANALY GEO",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE] <- 1
math1111$CC_CALC_I_00[grepl("ANL GEO",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE] <- 1
math1111$CC_CALC_I_00[grepl("MATH 181",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE] <- 1
math1111$CC_CALC_I_00[math1111$CC_CALC_II_00==1] <- 0
math1111$CC_CALC_I_00[math1111$CC_PRE_CALC_00==1] <- 0

table(math1111$CC_CALC_I_00)

#trig
math1111$CC_TRIG_00 <- 0
math1111$CC_TRIG_00[grepl("trig",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE] <- 1
math1111$CC_TRIG_00[math1111$CC_COLL_ALG_00==1] <- 0
math1111$CC_TRIG_00[math1111$CC_PRE_CALC_00==1] <- 0

table(math1111$CC_TRIG_00)

#discrete math
math1111$CC_DISCRETE_MATH_00 <- 0
math1111$CC_DISCRETE_MATH_00[grepl("DISCRETE",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE] <- 1

table(math1111$CC_DISCRETE_MATH_00)

#finite math
math1111$cc_finite_00 <- 0
math1111$cc_finite_00[grepl("FINITE",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE] <- 1
```

```
table(math1111$cc_finite,math1111$cc_finite_00)

#differential equations
math1111$CC_DIFF_EQ_00 <- 0
math1111$CC_DIFF_EQ_00[grepl("DIFF",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE] <- 1
math1111$CC_DIFF_EQ_00[grepl("LINEAR",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE] <- 1
math1111$CC_DIFF_EQ_00[grepl("LIN ALG",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE] <- 1

table(math1111$CC_DIFF_EQ_00)

#transfer level rank
math1111$CC_TRANSFER_MATH_TYPE_00 <- 0
math1111$CC_TRANSFER_MATH_TYPE_00[math1111$CC_STATISTICS_00==1] <- 1
math1111$CC_TRANSFER_MATH_TYPE_00[math1111$CC_LA_MATH_00==1] <- 2
math1111$CC_TRANSFER_MATH_TYPE_00[math1111$cc_finite_00==1] <- 3
math1111$CC_TRANSFER_MATH_TYPE_00[math1111$CC_COLL_ALG_00==1] <- 4
math1111$CC_TRANSFER_MATH_TYPE_00[math1111$CC_BUS_MATH_00==1] <- 5
math1111$CC_TRANSFER_MATH_TYPE_00[math1111$CC_TRIG_00==1] <- 6
math1111$CC_TRANSFER_MATH_TYPE_00[math1111$CC_PRE_CALC_00==1] <- 7
math1111$CC_TRANSFER_MATH_TYPE_00[math1111$CC_DISCRETE_MATH_00==1] <- 8
math1111$CC_TRANSFER_MATH_TYPE_00[math1111$CC_CALC_I_00==1] <- 9
math1111$CC_TRANSFER_MATH_TYPE_00[math1111$CC_CALC_II_00==1] <- 10
math1111$CC_TRANSFER_MATH_TYPE_00[math1111$CC_DIFF_EQ_00==1] <- 11
math1111$CC_TRANSFER_MATH_TYPE_00[is.na(math1111$CC_00_COURSE_SUCCESS_IND)] <- NA

table(math1111$CC_TRANSFER_MATH_TYPE_00)

#Create SLAM (stats & lib arts math) and BSTEM flags
math1111$CC_SLAM <- 0
math1111$CC_SLAM[math1111$CC_STATISTICS_00==1 | math1111$CC_LA_MATH_00==1 | math1111$cc_finite_00==1] <- 1
math1111$CC_BSTEM <- 0
math1111$CC_BSTEM[math1111$CC_COLL_ALG_00==1 | math1111$CC_BUS_MATH_00==1 | math1111$CC_TRIG_00==1 |
math1111$CC_PRE_CALC_00==1 | math1111$CC_DISCRETE_MATH_00==1 | math1111$CC_CALC_I_00==1 | math1111$CC_CALC_
II_00==1 | math1111$CC_DIFF_EQ_00==1] <- 1
#tags only the first level of entry into BTEM (omits calculus)
math1111$CC_BSTEMentrylevel <- 0
math1111$CC_BSTEMentrylevel[math1111$CC_COLL_ALG_00==1 | math1111$CC_BUS_MATH_00==1 | math1111$CC_TRIG_00==1 |
math1111$CC_PRE_CALC_00==1 | math1111$CC_DISCRETE_MATH_00==1] <- 1

table(math1111$CC_SLAM,math1111$CC_BSTEM)
prop.table(table(math1111$CC_SLAM,math1111$CC_BSTEM))

#SLAM v BTEM flag
math1111$CC_SLAM_v_BSTEM[math1111$CC_SLAM==1] <- 0
math1111$CC_SLAM_v_BSTEM[math1111$CC_BSTEM==1] <- 1

table(math1111$CC_SLAM_v_BSTEM)
prop.table(table(math1111$CC_SLAM_v_BSTEM))
table(math1111$CC_BSTEM,math1111$CC_PRE_CALC_00)
prop.table(table(math1111$CC_BSTEM,math1111$CC_PRE_CALC_00),1)
#estimate the proportion of students using precalc as transfer level bstem entry
table(math1111$CC_BSTEMentrylevel,math1111$CC_PRE_CALC_00)
prop.table(table(math1111$CC_BSTEMentrylevel,math1111$CC_PRE_CALC_00),1)
```

```
#One level below math nonstem variable
math1111$CC_NON_STEM_A <- 0
math1111$CC_NON_STEM_A[
        (
         grepl("stat",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE |
         grepl("DATA ANALYSIS",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE |
         grepl("TECHNICAL",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE |
         grepl("NON-SCIENCE",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE |
         grepl("LIB ART",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE |
         grepl("WLD",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE |
         grepl("OFFER",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE |
         grepl("PRACTICAL",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE |
         grepl("CONTEMPORARY CAREERS",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE |
         grepl("SURVEY OF MATH",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE |
         grepl("LITERACY",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE |
         grepl("INDUST",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE |
         grepl("INT ALGEBRA W/ APPLICATIONS",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE |
         grepl("GENERAL EDUCATION ALGEBRA",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE |
         grepl("HEALTH",math1111$CC_00_COURSE_TITLE,ignore.case=TRUE)==TRUE
        )
        & math1111$CC_FIRST_COURSE_LEVEL_ID=="A"
       ] <- 1

table(math1111$CC_NON_STEM_A)
```

Appendix B: R Code for Regression Analyses

Transfer-Level English R Code

```
#regression not filtered for GPA for all node use
e1 <- CC_FIRST_COURSE_SUCCESS_IND ~ HS_11_GPA_CUM + ACCUPLACER_ENGL_SENTENCE_SKILLS_SCORE +
ACCUPLACER_ENGL_READING_COMP_SCORE
e1lin2 <- lm(e1,data=engl1111[engl1111$CC_FIRST_COURSE_LEVEL_ID==”Y”,])
summary(e1lin2)
#node 1
tapply(engl1111$HS_LAST_GPA_CUM[(engl1111$ACCUPLACER_ENGL_READING_COMP_SCORE>=0 & engl1111$ACCUPLACER_ENGL_
SENTENCE_SKILLS_SCORE>=0 & engl1111$HS_LAST_GPA_CUM < 1.9)],engl1111$CC_FIRST_LEVEL_RANK[(engl1111$ACCUPLACER_
ENGL_READING_COMP_SCORE>=0 & engl1111$ACCUPLACER_ENGL_SENTENCE_SKILLS_SCORE>=0 & engl1111$HS_LAST_GPA_CUM <
1.9)],mean)
tapply(engl1111$ACCUPLACER_ENGL_SENTENCE_SKILLS_SCORE[(engl1111$ACCUPLACER_ENGL_SENTENCE_SKILLS_SCORE>=0 &
engl1111$HS_LAST_GPA_CUM < 1.9)],engl1111$CC_FIRST_LEVEL_RANK[(engl1111
$ACCUPLACER_ENGL_SENTENCE_SKILLS_SCORE>=0 & engl1111$HS_LAST_GPA_CUM < 1.9)],mean)
tapply(engl1111$ACCUPLACER_ENGL_READING_COMP_SCORE[(engl1111$ACCUPLACER_ENGL_READING_COMP_SCORE>=0 &
engl1111$HS_LAST_GPA_CUM < 1.9)],engl1111$CC_FIRST_LEVEL_RANK[(engl1111$ACCUPLACER_ENGL_READING_COMP_SCORE>=0 &
engl1111$HS_LAST_GPA_CUM < 1.9)],mean)
#table(engl1111$CC_FIRST_LEVEL_RANK[engl1111$HS_11_GPA_CUM < 1.9 & engl1111$CC_00_COURSE_LEVEL_ID==”Y”])
table(engl1111$CC_FIRST_LEVEL_RANK[engl1111$HS_11_GPA_CUM < 1.9])
englpred0.1 <- data.frame(HS_11_GPA_CUM=1.62,ACCUPLACER_ENGL_SENTENCE_SKILLS_SCORE=95.6,ACCUPLACER_ENGL_
READING_COMP_SCORE=88.7)
predict(e1lin2,englpred0.1,se.fit = TRUE,interval=”confidence”)
englpred1.1 <- data.frame(HS_11_GPA_CUM=1.61,ACCUPLACER_ENGL_SENTENCE_SKILLS_SCORE=79.1,ACCUPLACER_ENGL_
READING_COMP_SCORE=73.2)
predict(e1lin2,englpred1.1,se.fit = TRUE,interval=”confidence”)
englpred2.1 <- data.frame(HS_11_GPA_CUM=1.59,ACCUPLACER_ENGL_SENTENCE_SKILLS_SCORE=67.7,ACCUPLACER_ENGL_
READING_COMP_SCORE=61.1)
predict(e1lin2,englpred2.1,se.fit = TRUE,interval=”confidence”)
englpred3.1 <- data.frame(HS_11_GPA_CUM=1.58,ACCUPLACER_ENGL_SENTENCE_SKILLS_SCORE=61.2,ACCUPLACER_ENGL_
READING_COMP_SCORE=54.7)
predict(e1lin2,englpred3.1,se.fit = TRUE,interval=”confidence”)
englpred4.1 <- data.frame(HS_11_GPA_CUM=1.60,ACCUPLACER_ENGL_SENTENCE_SKILLS_SCORE=59.4,ACCUPLACER_ENGL_
READING_COMP_SCORE=51.4)
predict(e1lin2,englpred4.1,se.fit = TRUE,interval=”confidence”)
#node 2
tapply(engl1111$HS_LAST_GPA_CUM[(engl1111$ACCUPLACER_ENGL_READING_COMP_SCORE>=0 & engl1111$ACCUPLACER_ENGL_
SENTENCE_SKILLS_SCORE>=0 & engl1111$HS_LAST_GPA_CUM >= 1.9 & engl1111$HS_LAST_GPA_CUM < 2.6 & engl1111$HS_11_
COURSE_GRADE_POINTS < 1.7)],engl1111$CC_FIRST_LEVEL_RANK[(engl1111$ACCUPLACER_ENGL_READING_COMP_SCORE>=0 &
engl1111$ACCUPLACER_ENGL_SENTENCE_SKILLS_SCORE>=0 & engl1111$HS_LAST_GPA_CUM >= 1.9 & engl1111$HS_LAST_GPA_CUM
< 2.6 & engl1111$HS_11_COURSE_GRADE_POINTS < 1.7)],mean)
tapply(engl1111$ACCUPLACER_ENGL_SENTENCE_SKILLS_SCORE[(engl1111$ACCUPLACER_ENGL_SENTENCE_SKILLS_SCORE>=0
& engl1111$HS_LAST_GPA_CUM >= 1.9 & engl1111$HS_LAST_GPA_CUM < 2.6 & engl1111$HS_11_COURSE_GRADE_POINTS <
1.7)],engl1111$CC_FIRST_LEVEL_RANK[(engl1111$ACCUPLACER_ENGL_SENTENCE_SKILLS_SCORE>=0 & engl1111$HS_LAST_GPA_
CUM >= 1.9 & engl1111$HS_LAST_GPA_CUM < 2.6 & engl1111$HS_11_COURSE_GRADE_POINTS < 1.7)],mean)
tapply(engl1111$ACCUPLACER_ENGL_READING_COMP_SCORE[(engl1111$ACCUPLACER_ENGL_READING_COMP_SCORE>=0
& engl1111$HS_LAST_GPA_CUM >= 1.9 & engl1111$HS_LAST_GPA_CUM < 2.6 & engl1111$HS_11_COURSE_GRADE_POINTS <
1.7)],engl1111$CC_FIRST_LEVEL_RANK[(engl1111$ACCUPLACER_ENGL_READING_COMP_SCORE>=0 & engl1111$HS_LAST_GPA_CUM
>= 1.9 & engl1111$HS_LAST_GPA_CUM < 2.6 & engl1111$HS_11_COURSE_GRADE_POINTS < 1.7)],mean)
table(engl1111$CC_FIRST_LEVEL_RANK[engl1111$HS_11_GPA_CUM >= 1.9 & engl1111$HS_LAST_GPA_CUM < 2.6 & engl1111$HS_11_
COURSE_GRADE_POINTS < 1.7])
englpred0.2 <- data.frame(HS_11_GPA_CUM=2.42,ACCUPLACER_ENGL_SENTENCE_SKILLS_SCORE=98.1,ACCUPLACER_ENGL_
READING_COMP_SCORE=90.2)
predict(e1lin2,englpred0.2,se.fit = TRUE,interval=”confidence”)
englpred1.2 <- data.frame(HS_11_GPA_CUM=2.20,ACCUPLACER_ENGL_SENTENCE_SKILLS_SCORE=81.8,ACCUPLACER_ENGL_
READING_COMP_SCORE=74.8)
predict(e1lin2,englpred1.2,se.fit = TRUE,interval=”confidence”)
englpred2.2 <- data.frame(HS_11_GPA_CUM=2.20,ACCUPLACER_ENGL_SENTENCE_SKILLS_SCORE=70.4,ACCUPLACER_ENGL_
READING_COMP_SCORE=63.1)
predict(e1lin2,englpred2.2,se.fit = TRUE,interval=”confidence”)
englpred3.2 <- data.frame(HS_11_GPA_CUM=2.18,ACCUPLACER_ENGL_SENTENCE_SKILLS_SCORE=62.7,ACCUPLACER_ENGL_
READING_COMP_SCORE=55.3)
```

```
predict(e1lin2,englpred3.2,se.fit = TRUE,interval=”confidence”)
englpred4.2 <- data.frame(HS_11_GPA_CUM=2.18,ACCUPLACER_ENGL_SENTENCE_SKILLS_SCORE=60.0,ACCUPLACER_ENGL_
READING_COMP_SCORE=50.6)
predict(e1lin2,englpred4.2,se.fit = TRUE,interval=”confidence”)
#node 3
tapply(engl1111$HS_LAST_GPA_CUM[(engl1111$ACCUPLACER_ENGL_READING_COMP_SCORE>=0 & engl1111$ACCUPLACER_ENGL_
SENTENCE_SKILLS_SCORE>=0 & engl1111$HS_LAST_GPA_CUM >= 1.9 & engl1111$HS_LAST_GPA_CUM < 2.6 & engl1111$HS_11_
COURSE_GRADE_POINTS >= 1.7)],engl1111$CC_FIRST_LEVEL_RANK[(engl1111$ACCUPLACER_ENGL_READING_COMP_SCORE>=0 &
engl1111$ACCUPLACER_ENGL_SENTENCE_SKILLS_SCORE>=0 & engl1111$HS_LAST_GPA_CUM >= 1.9 & engl1111$HS_LAST_GPA_CUM
< 2.6 & engl1111$HS_11_COURSE_GRADE_POINTS >= 1.7)],mean)
tapply(engl1111$ACCUPLACER_ENGL_SENTENCE_SKILLS_SCORE[(engl1111$ACCUPLACER_ENGL_SENTENCE_SKILLS_SCORE>=0
& engl1111$HS_LAST_GPA_CUM >= 1.9 & engl1111$HS_LAST_GPA_CUM < 2.6 & engl1111$HS_11_COURSE_GRADE_POINTS >=
1.7)],engl1111$CC_FIRST_LEVEL_RANK[(engl1111$ACCUPLACER_ENGL_SENTENCE_SKILLS_SCORE>=0 & engl1111$HS_LAST_GPA_
CUM >= 1.9 & engl1111$HS_LAST_GPA_CUM < 2.6 & engl1111$HS_11_COURSE_GRADE_POINTS >= 1.7)],mean)
tapply(engl1111$ACCUPLACER_ENGL_READING_COMP_SCORE[(engl1111$ACCUPLACER_ENGL_READING_COMP_SCORE>=0
& engl1111$HS_LAST_GPA_CUM >= 1.9 & engl1111$HS_LAST_GPA_CUM < 2.6 & engl1111$HS_11_COURSE_GRADE_POINTS >=
1.7)],engl1111$CC_FIRST_LEVEL_RANK[(engl1111$ACCUPLACER_ENGL_READING_COMP_SCORE>=0 & engl1111$HS_LAST_GPA_CUM
>= 1.9 & engl1111$HS_LAST_GPA_CUM < 2.6 & engl1111$HS_11_COURSE_GRADE_POINTS >= 1.7)],mean)
table(engl1111$CC_FIRST_LEVEL_RANK[engl1111$HS_11_GPA_CUM >= 1.9 & engl1111$HS_LAST_GPA_CUM < 2.6 & engl1111$HS_11_
COURSE_GRADE_POINTS >= 1.7])
englpred0.3 <- data.frame(HS_11_GPA_CUM=2.32,ACCUPLACER_ENGL_SENTENCE_SKILLS_SCORE=97.5,ACCUPLACER_ENGL_
READING_COMP_SCORE=89.2)
predict(e1lin2,englpred0.3,se.fit = TRUE,interval=”confidence”)
englpred1.3 <- data.frame(HS_11_GPA_CUM=2.29,ACCUPLACER_ENGL_SENTENCE_SKILLS_SCORE=81.7,ACCUPLACER_ENGL_
READING_COMP_SCORE=74.1)
predict(e1lin2,englpred1.3,se.fit = TRUE,interval=”confidence”)
englpred2.3 <- data.frame(HS_11_GPA_CUM=2.27,ACCUPLACER_ENGL_SENTENCE_SKILLS_SCORE=69.7,ACCUPLACER_ENGL_
READING_COMP_SCORE=62.7)
predict(e1lin2,englpred2.3,se.fit = TRUE,interval=”confidence”)
englpred3.3 <- data.frame(HS_11_GPA_CUM=2.28,ACCUPLACER_ENGL_SENTENCE_SKILLS_SCORE=64.2,ACCUPLACER_ENGL_
READING_COMP_SCORE=55.8)
predict(e1lin2,englpred3.3,se.fit = TRUE,interval=”confidence”)
englpred4.3 <- data.frame(HS_11_GPA_CUM=2.26,ACCUPLACER_ENGL_SENTENCE_SKILLS_SCORE=59.8,ACCUPLACER_ENGL_
READING_COMP_SCORE=53.1)
predict(e1lin2,englpred4.3,se.fit = TRUE,interval=”confidence”)
#node 4
tapply(engl1111$HS_LAST_GPA_CUM[(engl1111$ACCUPLACER_ENGL_READING_COMP_SCORE>=0 & engl1111$ACCUPLACER_
ENGL_SENTENCE_SKILLS_SCORE>=0 & engl1111$HS_LAST_GPA_CUM >= 2.6 & engl1111$HS_11_COURSE_GRADE_POINTS >=
1.7)],engl1111$CC_FIRST_LEVEL_RANK[(engl1111$ACCUPLACER_ENGL_READING_COMP_SCORE>=0 & engl1111$ACCUPLACER_ENGL_
SENTENCE_SKILLS_SCORE>=0 & engl1111$HS_LAST_GPA_CUM >= 2.6 & engl1111$HS_11_COURSE_GRADE_POINTS >= 1.7)],mean)
tapply(engl1111$ACCUPLACER_ENGL_SENTENCE_SKILLS_SCORE[(engl1111$ACCUPLACER_ENGL_SENTENCE_SKILLS_SCORE>=0
& engl1111$HS_LAST_GPA_CUM >= 2.6 & engl1111$HS_11_COURSE_GRADE_POINTS >= 1.7)],engl1111$CC_FIRST_LEVEL_
RANK[(engl1111$ACCUPLACER_ENGL_SENTENCE_SKILLS_SCORE>=0 & engl1111$HS_LAST_GPA_CUM >= 2.6 & engl1111$HS_11_
COURSE_GRADE_POINTS >= 1.7)],mean)
tapply(engl1111$ACCUPLACER_ENGL_READING_COMP_SCORE[(engl1111$ACCUPLACER_ENGL_READING_COMP_SCORE>=0
& engl1111$HS_LAST_GPA_CUM >= 2.6 & engl1111$HS_11_COURSE_GRADE_POINTS >= 1.7)],engl1111$CC_FIRST_LEVEL_
RANK[(engl1111$ACCUPLACER_ENGL_READING_COMP_SCORE>=0 & engl1111$HS_LAST_GPA_CUM >= 2.6 & engl1111$HS_11_
COURSE_GRADE_POINTS >= 1.7)],mean)
table(engl1111$CC_FIRST_LEVEL_RANK[engl1111$HS_11_GPA_CUM >= 2.6 & engl1111$HS_11_COURSE_GRADE_
POINTS >= 1.7])
englpred0.4 <- data.frame(HS_11_GPA_CUM=3.11,ACCUPLACER_ENGL_SENTENCE_SKILLS_SCORE=101.2,ACCUPLACER_ENGL_
READING_COMP_SCORE=91.9)
predict(e1lin2,englpred0.4,se.fit = TRUE,interval=”confidence”)
englpred1.4 <- data.frame(HS_11_GPA_CUM=3.01,ACCUPLACER_ENGL_SENTENCE_SKILLS_SCORE=84.3,ACCUPLACER_ENGL_
READING_COMP_SCORE=76.4)
predict(e1lin2,englpred1.4,se.fit = TRUE,interval=”confidence”)
englpred2.4 <- data.frame(HS_11_GPA_CUM=2.96,ACCUPLACER_ENGL_SENTENCE_SKILLS_SCORE=72.4,ACCUPLACER_ENGL_
READING_COMP_SCORE=64.3)
predict(e1lin2,englpred2.4,se.fit = TRUE,interval=”confidence”)
englpred3.4 <- data.frame(HS_11_GPA_CUM=2.94,ACCUPLACER_ENGL_SENTENCE_SKILLS_SCORE=64.8,ACCUPLACER_ENGL_
READING_COMP_SCORE=55.5)
predict(e1lin2,englpred3.4,se.fit = TRUE,interval=”confidence”)
englpred4.4 <- data.frame(HS_11_GPA_CUM=2.93,ACCUPLACER_ENGL_SENTENCE_SKILLS_SCORE=60.0,ACCUPLACER_ENGL_
READING_COMP_SCORE=49.2)
predict(e1lin2,englpred4.4,se.fit = TRUE,interval=”confidence”)
```

Statistics R Code

```
f5 <- CC_FIRST_COURSE_SUCCESS_IND ~ HS_11_GPA_CUM + ACCUPLACER_MATH_COLLEGE_SCORE
lr5.2 <- glm(f5,family = “binomial”,data=math1111[math1111$CC_FIRST_COURSE_LEVEL_ID==”Y” & math1111$CC_STATISTICS==1,])
summary(lr5.2)
#statistics node 1
tapply(math1111$HS_11_GPA_CUM[math1111$ACCUPLACER_MATH_ELEM_ALG_SCORE >= 0 & math1111$HS_11_GPA_CUM <
2.3],math1111$CC_FIRST_LEVEL_RANK[math1111$ACCUPLACER_MATH_ELEM_ALG_SCORE >=0 & math1111$HS_11_GPA_CUM <
2.3],mean)
tapply(math1111$ACCUPLACER_MATH_COLLEGE_SCORE[math1111$ACCUPLACER_MATH_COLLEGE_SCORE >= 0],math1111$CC_
FIRST_LEVEL_RANK[math1111$ACCUPLACER_MATH_COLLEGE_SCORE >= 0],mean)
table(math1111$CC_FIRST_LEVEL_RANK[math1111$HS_11_GPA_CUM < 2.3])
statpred0 <- data.frame(HS_11_GPA_CUM=1.98,ACCUPLACER_MATH_COLLEGE_SCORE=58.3)
predict(lrlin5.2,statpred0,se.fit = TRUE,interval=”confidence”)
statpred1 <- data.frame(HS_11_GPA_CUM=1.92,ACCUPLACER_MATH_COLLEGE_SCORE=36.1)
predict(lrlin5.2,statpred1,se.fit = TRUE,interval=”confidence”)
statpred2 <- data.frame(HS_11_GPA_CUM=1.79,ACCUPLACER_MATH_COLLEGE_SCORE=27.3)
predict(lrlin5.2,statpred2,se.fit = TRUE,interval=”confidence”)
statpred3 <- data.frame(HS_11_GPA_CUM=1.74,ACCUPLACER_MATH_COLLEGE_SCORE=27.9)
predict(lrlin5.2,statpred3,se.fit = TRUE,interval=”confidence”)
statpred4 <- data.frame(HS_11_GPA_CUM=1.72,ACCUPLACER_MATH_COLLEGE_SCORE=25)
predict(lrlin5.2,statpred4,se.fit = TRUE,interval=”confidence”)
#statistics node 2
tapply(math1111$HS_11_GPA_CUM[math1111$ACCUPLACER_MATH_ELEM_ALG_SCORE >= 0 & math1111$HS_11_GPA_CUM >= 2.3 &
math1111$HS_11_GPA_CUM < 3 & math1111$ALG_I_UP11_C==0 & math1111$PRE_CALC_UP11_C==0],math1111$CC_FIRST_LEVEL_
RANK[math1111$ACCUPLACER_MATH_ELEM_ALG_SCORE >=0 & math1111$HS_11_GPA_CUM >= 2.3 & math1111$HS_11_GPA_CUM < 3
& math1111$ALG_I_UP11_C==0 & math1111$PRE_CALC_UP11_C==0],mean)
table(math1111$CC_FIRST_LEVEL_RANK[math1111$HS_11_GPA_CUM >= 2.3 & math1111$HS_11_GPA_CUM < 3 & math1111$ALG_I_
UP11_C==0 & math1111$PRE_CALC_UP11_C==0])
statpred0.2 <- data.frame(HS_11_GPA_CUM=2.67,ACCUPLACER_MATH_COLLEGE_SCORE=58.3)
predict(lrlin5.2,statpred0.2,se.fit = TRUE,interval=”confidence”)
statpred1.2 <- data.frame(HS_11_GPA_CUM=2.64,ACCUPLACER_MATH_COLLEGE_SCORE=36.1)
predict(lrlin5.2,statpred1.2,se.fit = TRUE,interval=”confidence”)
statpred2.2 <- data.frame(HS_11_GPA_CUM=2.59,ACCUPLACER_MATH_COLLEGE_SCORE=27.3)
predict(lrlin5.2,statpred2.2,se.fit = TRUE,interval=”confidence”)
statpred3.2 <- data.frame(HS_11_GPA_CUM=2.56,ACCUPLACER_MATH_COLLEGE_SCORE=27.9)
predict(lrlin5.2,statpred3.2,se.fit = TRUE,interval=”confidence”)
statpred4.2 <- data.frame(HS_11_GPA_CUM=2.56,ACCUPLACER_MATH_COLLEGE_SCORE=25)
predict(lrlin5.2,statpred4.2,se.fit = TRUE,interval=”confidence”)
#statistics node 3
tapply(math1111$HS_11_GPA_CUM[math1111$ACCUPLACER_MATH_ELEM_ALG_SCORE >= 0 & math1111$HS_11_GPA_CUM >= 2.3 &
math1111$HS_11_GPA_CUM < 3 & math1111$ALG_I_UP11_C==1 & math1111$PRE_CALC_UP11_C==0],math1111$CC_FIRST_LEVEL_
RANK[math1111$ACCUPLACER_MATH_ELEM_ALG_SCORE >=0 & math1111$HS_11_GPA_CUM >= 2.3 & math1111$HS_11_GPA_CUM < 3
& math1111$ALG_I_UP11_C==1 & math1111$PRE_CALC_UP11_C==0],mean)
table(math1111$CC_FIRST_LEVEL_RANK[math1111$HS_11_GPA_CUM >= 2.3 & math1111$HS_11_GPA_CUM < 3 & math1111$ALG_I_
UP11_C==1 & math1111$PRE_CALC_UP11_C==0])
statpred0.3 <- data.frame(HS_11_GPA_CUM=2.68,ACCUPLACER_MATH_COLLEGE_SCORE=58.3)
predict(lrlin5.2,statpred0.3,se.fit = TRUE,interval=”confidence”)
statpred1.3 <- data.frame(HS_11_GPA_CUM=2.64,ACCUPLACER_MATH_COLLEGE_SCORE=36.1)
predict(lrlin5.2,statpred1.3,se.fit = TRUE,interval=”confidence”)
statpred2.3 <- data.frame(HS_11_GPA_CUM=2.61,ACCUPLACER_MATH_COLLEGE_SCORE=27.3)
predict(lrlin5.2,statpred2.3,se.fit = TRUE,interval=”confidence”)
statpred3.3 <- data.frame(HS_11_GPA_CUM=2.59,ACCUPLACER_MATH_COLLEGE_SCORE=27.9)
predict(lrlin5.2,statpred3.3,se.fit = TRUE,interval=”confidence”)
statpred4.3 <- data.frame(HS_11_GPA_CUM=2.59,ACCUPLACER_MATH_COLLEGE_SCORE=25)
predict(lrlin5.2,statpred4.3,se.fit = TRUE,interval=”confidence”)
#statistics node 4
tapply(math1111$HS_11_GPA_CUM[math1111$ACCUPLACER_MATH_ELEM_ALG_SCORE >= 0 & math1111$HS_11_GPA_CUM >= 2.3 &
math1111$HS_11_GPA_CUM < 3 & math1111$PRE_CALC_UP11_C==1],math1111$CC_FIRST_LEVEL_RANK[math1111$ACCUPLACER_
MATH_ELEM_ALG_SCORE >=0 & math1111$HS_11_GPA_CUM >= 2.3 & math1111$HS_11_GPA_CUM < 3 & math1111$PRE_CALC_UP11_
C==1],mean)
table(math1111$CC_FIRST_LEVEL_RANK[math1111$HS_11_GPA_CUM >= 2.3 & math1111$HS_11_GPA_CUM < 3 & math1111$PRE_
CALC_UP11_C==1])
statpred0.4 <- data.frame(HS_11_GPA_CUM=2.74,ACCUPLACER_MATH_COLLEGE_SCORE=58.3)
predict(lrlin5.2,statpred0.4,se.fit = TRUE,interval=”confidence”)
statpred1.4 <- data.frame(HS_11_GPA_CUM=2.73,ACCUPLACER_MATH_COLLEGE_SCORE=36.1)
predict(lrlin5.2,statpred1.4,se.fit = TRUE,interval=”confidence”)
```

```
statpred2.4 <- data.frame(HS_11_GPA_CUM=2.73,ACCUPLACER_MATH_COLLEGE_SCORE=27.3)
predict(lrlin5.2,statpred2.4,se.fit = TRUE,interval="confidence")
statpred3.4 <- data.frame(HS_11_GPA_CUM=2.67,ACCUPLACER_MATH_COLLEGE_SCORE=27.9)
predict(lrlin5.2,statpred3.4,se.fit = TRUE,interval="confidence")
statpred4.4 <- data.frame(HS_11_GPA_CUM=2.70,ACCUPLACER_MATH_COLLEGE_SCORE=25)
predict(lrlin5.2,statpred4.4,se.fit = TRUE,interval="confidence")
#statistics node 5
tapply(math1111$HS_11_GPA_CUM[math1111$ACCUPLACER_MATH_ELEM_ALG_SCORE >= 0 & math1111$HS_11_GPA_CUM >= 3],math1111$CC_FIRST_LEVEL_RANK[math1111$ACCUPLACER_MATH_ELEM_ALG_SCORE >=0 & math1111$HS_11_GPA_CUM >= 3],mean)
table(math1111$CC_FIRST_LEVEL_RANK[math1111$HS_11_GPA_CUM >= 3])
statpred0.5 <- data.frame(HS_11_GPA_CUM=3.39,ACCUPLACER_MATH_COLLEGE_SCORE=58.3)
predict(lrlin5.2,statpred0.5,se.fit = TRUE,interval="confidence")
statpred1.5 <- data.frame(HS_11_GPA_CUM=3.30,ACCUPLACER_MATH_COLLEGE_SCORE=36.1)
predict(lrlin5.2,statpred1.5,se.fit = TRUE,interval="confidence")
statpred2.5 <- data.frame(HS_11_GPA_CUM=3.24,ACCUPLACER_MATH_COLLEGE_SCORE=27.3)
predict(lrlin5.2,statpred2.5,se.fit = TRUE,interval="confidence")
statpred3.5 <- data.frame(HS_11_GPA_CUM=3.25,ACCUPLACER_MATH_COLLEGE_SCORE=27.9)
predict(lrlin5.2,statpred3.5,se.fit = TRUE,interval="confidence")
statpred4.5 <- data.frame(HS_11_GPA_CUM=3.22,ACCUPLACER_MATH_COLLEGE_SCORE=25)
predict(lrlin5.2,statpred4.5,se.fit = TRUE,interval="confidence")
```

Pre-calculus R Code

```
f5 <- CC_FIRST_COURSE_SUCCESS_IND ~ HS_11_GPA_CUM + ACCUPLACER_MATH_COLLEGE_SCORE
lrlin5.1 <- lm(f5,data=math1111[math1111$CC_FIRST_COURSE_LEVEL_ID=="Y" & math1111$CC_PRE_CALC==1,])
summary(lrlin5.1)
tapply(math1111$HS_11_GPA_CUM[math1111$ACCUPLACER_MATH_COLLEGE_SCORE >= 0 & math1111$HS_11_GPA_CUM < 2.6 & math1111$PRE_CALC_UP11==0],math1111$CC_FIRST_LEVEL_RANK[math1111$ACCUPLACER_MATH_COLLEGE_SCORE >= 0 & math1111$HS_11_GPA_CUM < 2.6 & math1111$PRE_CALC_UP11==0],mean)
table(math1111$CC_FIRST_LEVEL_RANK[math1111$HS_11_GPA_CUM < 2.6 & math1111$PRE_CALC_UP11==0])
precalcpred0.1 <- data.frame(HS_11_GPA_CUM=2.23,ACCUPLACER_MATH_COLLEGE_SCORE=58.3)
predict(lrlin5.1,precalcpred0.1,se.fit = TRUE,interval="confidence")
precalcpred1.1 <- data.frame(HS_11_GPA_CUM=2.14,ACCUPLACER_MATH_COLLEGE_SCORE=36.1)
predict(lrlin5.1,precalcpred1.1,se.fit = TRUE,interval="confidence")
precalcpred2.1 <- data.frame(HS_11_GPA_CUM=2.03,ACCUPLACER_MATH_COLLEGE_SCORE=27.3)
predict(lrlin5.1,precalcpred2.1,se.fit = TRUE,interval="confidence")
precalcpred3.1 <- data.frame(HS_11_GPA_CUM=2.02,ACCUPLACER_MATH_COLLEGE_SCORE=27.9)
predict(lrlin5.1,precalcpred3.1,se.fit = TRUE,interval="confidence")
precalcpred4.1 <- data.frame(HS_11_GPA_CUM=2.05,ACCUPLACER_MATH_COLLEGE_SCORE=25)
predict(lrlin5.1,precalcpred4.1,se.fit = TRUE,interval="confidence")
#precalculus node 2
tapply(math1111$HS_11_GPA_CUM[math1111$ACCUPLACER_MATH_COLLEGE_SCORE >= 0 & math1111$HS_11_GPA_CUM < 2.6 & math1111$PRE_CALC_UP11==1],math1111$CC_FIRST_LEVEL_RANK[math1111$ACCUPLACER_MATH_COLLEGE_SCORE >= 0 & math1111$HS_11_GPA_CUM < 2.6 & math1111$PRE_CALC_UP11==1],mean)
table(math1111$CC_FIRST_LEVEL_RANK[math1111$HS_11_GPA_CUM < 2.6 & math1111$PRE_CALC_UP11==1])
precalcpred0.2 <- data.frame(HS_11_GPA_CUM=2.28,ACCUPLACER_MATH_COLLEGE_SCORE=58.3)
predict(lrlin5.1,precalcpred0.2,se.fit = TRUE,interval="confidence")
precalcpred1.2 <- data.frame(HS_11_GPA_CUM=2.28,ACCUPLACER_MATH_COLLEGE_SCORE=36.1)
predict(lrlin5.1,precalcpred1.2,se.fit = TRUE,interval="confidence")
precalcpred2.2 <- data.frame(HS_11_GPA_CUM=2.31,ACCUPLACER_MATH_COLLEGE_SCORE=27.3)
predict(lrlin5.1,precalcpred2.2,se.fit = TRUE,interval="confidence")
precalcpred3.2 <- data.frame(HS_11_GPA_CUM=2.49,ACCUPLACER_MATH_COLLEGE_SCORE=27.9)
predict(lrlin5.1,precalcpred3.2,se.fit = TRUE,interval="confidence")
precalcpred4.2 <- data.frame(HS_11_GPA_CUM=2.32,ACCUPLACER_MATH_COLLEGE_SCORE=25)
predict(lrlin5.1,precalcpred4.2,se.fit = TRUE,interval="confidence")
#precalculus node 3
tapply(math1111$HS_11_GPA_CUM[math1111$ACCUPLACER_MATH_COLLEGE_SCORE >= 0 & math1111$HS_11_GPA_CUM >= 2.6 & math1111$HS_11_GPA_CUM < 3.1],math1111$CC_FIRST_LEVEL_RANK[math1111$ACCUPLACER_MATH_COLLEGE_SCORE >= 0 & math1111$HS_11_GPA_CUM >= 2.6 & math1111$HS_11_GPA_CUM < 3.1],mean)
table(math1111$CC_FIRST_LEVEL_RANK[math1111$HS_11_GPA_CUM >= 2.6 & math1111$HS_11_GPA_CUM < 3.1])
precalcpred0.3 <- data.frame(HS_11_GPA_CUM=2.87,ACCUPLACER_MATH_COLLEGE_SCORE=58.3)
predict(lrlin5.1,precalcpred0.3,se.fit = TRUE,interval="confidence")
precalcpred1.3 <- data.frame(HS_11_GPA_CUM=2.85,ACCUPLACER_MATH_COLLEGE_SCORE=36.1)
predict(lrlin5.1,precalcpred1.3,se.fit = TRUE,interval="confidence")
precalcpred2.3 <- data.frame(HS_11_GPA_CUM=2.82,ACCUPLACER_MATH_COLLEGE_SCORE=27.3)
```

```
predict(lrlin5.1,precalcpred2.3,se.fit = TRUE,interval="confidence")
precalcpred3.3 <- data.frame(HS_11_GPA_CUM=2.84,ACCUPLACER_MATH_COLLEGE_SCORE=27.9)
predict(lrlin5.1,precalcpred3.3,se.fit = TRUE,interval="confidence")
precalcpred4.3 <- data.frame(HS_11_GPA_CUM=2.83,ACCUPLACER_MATH_COLLEGE_SCORE=25)
predict(lrlin5.1,precalcpred4.3,se.fit = TRUE,interval="confidence")
#precalculus node 4
tapply(math1111$HS_11_GPA_CUM[math1111$ACCUPLACER_MATH_COLLEGE_SCORE >= 0 & math1111$HS_11_GPA_CUM >= 3.1
& math1111$HS_11_GPA_CUM < 3.4],math1111$CC_FIRST_LEVEL_RANK[math1111$ACCUPLACER_MATH_COLLEGE_SCORE >= 0 &
math1111$HS_11_GPA_CUM >= 3.1 & math1111$HS_11_GPA_CUM < 3.4],mean)
table(math1111$CC_FIRST_LEVEL_RANK[math1111$HS_11_GPA_CUM >= 3.1 & math1111$HS_11_GPA_CUM < 3.4])
precalcpred0.4 <- data.frame(HS_11_GPA_CUM=3.24,ACCUPLACER_MATH_COLLEGE_SCORE=58.3)
predict(lrlin5.1,precalcpred0.4,se.fit = TRUE,interval="confidence")
precalcpred1.4 <- data.frame(HS_11_GPA_CUM=3.23,ACCUPLACER_MATH_COLLEGE_SCORE=36.1)
predict(lrlin5.1,precalcpred1.4,se.fit = TRUE,interval="confidence")
precalcpred2.4 <- data.frame(HS_11_GPA_CUM=3.23,ACCUPLACER_MATH_COLLEGE_SCORE=27.3)
predict(lrlin5.1,precalcpred2.4,se.fit = TRUE,interval="confidence")
precalcpred3.4 <- data.frame(HS_11_GPA_CUM=3.21,ACCUPLACER_MATH_COLLEGE_SCORE=27.9)
predict(lrlin5.1,precalcpred3.4,se.fit = TRUE,interval="confidence")
precalcpred4.4 <- data.frame(HS_11_GPA_CUM=3.23,ACCUPLACER_MATH_COLLEGE_SCORE=25)
predict(lrlin5.1,precalcpred4.4,se.fit = TRUE,interval="confidence")
#precalculus node 5
tapply(math1111$HS_11_GPA_CUM[math1111$ACCUPLACER_MATH_COLLEGE_SCORE >= 0 & math1111$HS_11_GPA_CUM >=
3.4],math1111$CC_FIRST_LEVEL_RANK[math1111$ACCUPLACER_MATH_COLLEGE_SCORE >= 0 & math1111$HS_11_GPA_CUM >=
3.4],mean)
table(math1111$CC_FIRST_LEVEL_RANK[math1111$HS_11_GPA_CUM >= 3.4])
precalcpred0.5 <- data.frame(HS_11_GPA_CUM=3.64,ACCUPLACER_MATH_COLLEGE_SCORE=58.3)
predict(lrlin5.1,precalcpred0.5,se.fit = TRUE,interval="confidence")
precalcpred1.5 <- data.frame(HS_11_GPA_CUM=3.59,ACCUPLACER_MATH_COLLEGE_SCORE=36.1)
predict(lrlin5.1,precalcpred1.5,se.fit = TRUE,interval="confidence")
precalcpred2.5 <- data.frame(HS_11_GPA_CUM=3.56,ACCUPLACER_MATH_COLLEGE_SCORE=27.3)
predict(lrlin5.1,precalcpred2.5,se.fit = TRUE,interval="confidence")
precalcpred3.5 <- data.frame(HS_11_GPA_CUM=3.63,ACCUPLACER_MATH_COLLEGE_SCORE=27.9)
predict(lrlin5.1,precalcpred3.5,se.fit = TRUE,interval="confidence")
precalcpred4.5 <- data.frame(HS_11_GPA_CUM=3.58,ACCUPLACER_MATH_COLLEGE_SCORE=25)
predict(lrlin5.1,precalcpred4.5,se.fit = TRUE,interval="confidence")
```

Appendix C: Regression model and success rate calculations

TRANSFER-LEVEL ENGLISH LOWEST NODE (HSGPA < 1.9)									
English HSGPA < 1.9	**Global regression with no GPA filter**				**Level below transfer average value within variable**				
var	Estimate	se	Pr(>|z|)	Test Coeff	0	1	2	3	4
constant	1.59E-02	3.11E-02	0.61	0.01588					
HS_11_GPA_CUM	2.44E-01	5.88E-03	<2e-16***	0.24410	1.62	1.61	1.59	1.58	.160
ACCUPLACER_ENGL_ SENTENCE_SKILLS_SCORE	-1.18E-04	3.01E-04	0.695	-0.00012	95.6	79.1	67.7	61.2	59.4
ACCUPLACER_ENGL_ READING_COMP_SCORE	6.71E-05	2.83E-04	0.813	0.00007	88.7	73.2	61.1	54.7	51.4
df=15918									
Multiple R-squared: 0.1009									
Success Prediction plugged with means					40.6%	40.4%	40.4%	39.9%	40.2%
prediction, x, from r (confirmation)					0.406	0.404	0.400	0.398	0.403
x - se					0.399	0.396	0.389	0.385	0.389
x + se					0.413	0.413	0.411	0.411	0.416
se.fit					0.007	0.009	0.011	0.013	0.013
Success rel to transfer level prediction					1.000	0.995	0.985	0.984	0.992
Count	37,545				7,248	13,174	11,414	4,827	882
Count of predicted successes with rel adj					3,117	5,635	4,833	2,041	376
low est count of success					3,117	5,514	4,699	1,976	364
high est count of success					3,117	5,755	4,967	2,107	389
Adjusted success rate within level				42.6%	43%	43%	42%	42%	43%
low est success rate				41.7%	43%	42%	41%	41%	41%
high est success rate				43.5%	43%	44%	44%	44%	44%
	Node Success	Regression	low error	high error					
Success rate	43.0%	42.6%	-0.9%	0.9%					

STATISTICS LOWEST NODE (HSGPA < 2.3)									
Statistics_HS_11_GPA_CUM<2.3					**Level below transfer average value within variable**				
var	Estimate	se	Pr(>|z|)	Test Coeff	0	1	2	3	4
constant	-0.2408621	0.097761	0.01395*	-0.241					
HS_11_GPA_CUM	0.2511751	0.030952	1.8e-15***	0.251	1.98	1.92	1.79	1.74	1.72
ACCUPLACER_MATH_COLLEGE_SCORE	0.0028206	0.000905	0.00188**	0.003	58.3	36.1	27.3	27.9	25.0
df=808									
Multiple R-squared: 0.09959									
Success Prediction plugged with means					42.1%	34.2%	28.7%	27.4%	26.1%
prediction from r					0.421	0.343	0.286	0.275	0.262
x - se					0.383	0.305	0.242	0.230	0.216
x + se					0.459	0.381	0.329	0.320	0.308
se.fit					0.038	0.038	0.044	0.045	0.046
Success rel to transfer level prediction					1.000	0.815	0.679	0.653	0.622
Count HS_11_GPA_CUM<2.3	37,545				4,352	11,309	17,186	11,268	4,931
Count of predicted successes with relative adjustment					1,741	3,689	4,667	2,943	1,226
low est count of success					1,741	3,278	3,954	2,465	1,010
high est count of success					1,741	4,099	5,380	3,422	1,442
Adjusted success rate within level				29.1%	40%	33%	27%	26%	25%
low est success rate				25.4%	40%	29%	23%	22%	20%
high est success rate				32.8%	40%	36%	31%	30%	29%
	Node Success	Regression	low error	high error					
Success rate in Statistics for HSGPA<2.3	40.0%	29.1%	-3.7%	3.7%					

PRE-CALCULUS LOWEST NODE (HSGPA < 2.6 AND NO HS PRE-CALCULUS)									
Precalc HSGPA<2.6 and no HS precalc					**Level below transfer average value within variable**				
var	Estimate	se	Pr(>|z|)	Test Coeff	0	1	2	3	4
constant	-0.216182	0.111242	0.0524	-0.216					
HS_11_GPA_CUM	0.209336	0.032594	2.57e-10***	0.209	2.23	2.14	2.03	2.02	2.05
ACCUPLACER_MATH_ COLLEGE_SCORE	0.004126	0.001021	6.00e-05**	0.004	58.3	36.1	27.3	27.9	25.0
df=658									
Multiple R-squared: 0.08938									
Success Prediction plugged with means					49.1%	38.1%	32.2%	32.2%	31.6%
prediction from r					0.491	0.381	0.321	0.322	0.316
x - se					0.460	0.341	0.274	0.275	0.267
x + se					0.522	0.421	0.369	0.369	0.365
se.fit					0.031	0.040	0.047	0.047	0.049
Success rel to transfer level prediction					1.000	0.775	0.654	0.655	0.644
Count	68,999				7,764	17,918	23,239	14,126	5,952
Count of predicted successes with relative adjustment					2,950	5,278	5,779	3,517	1,456
low est count of success					2,950	4,725	4,925	3,000	1,232
high est count of success					2,950	5,831	6,632	4,032	1,680
Adjusted success rate within level				27.5%	38%	29%	25%	25%	24%
low est success rate				24.4%	38%	26%	21%	21%	21%
high est success rate				30.6%	38%	33%	29%	29%	28%
	Node Success	Regression	low error	high error					
Success rate in Statistics for HSGPA<2.3	38.0%	27.5%	-3.1%	3.1%					

Appendix D: Excel formulae used to calculate estimates of direct transfer-level placement success rates

TRANSFER-LEVEL ENGLISH IS SHOWN AS AN EXAMPLE. NOTE THAT COLUMNS H THROUGH K1 HAVE THE SAME FORMULAE AS G BUT THE COLUMNS WERE NARROWED TO FIT TO VIEW									
English HSGPA<1.9	**Global regression with no GPA filter**				**Level below transfer average value within variable**				
var	Estimate	se	Pr(>|z|)	Test Coeff	0	1	2	3	4
constant	0.01588	0.03112	0.61	=B3					
HS_11_GPA_CUM	0.2441	0.005878	<2e-16***	=B4	1.619012	1.606507	1.587685	1.58437	1.59841
ACCUPLACER_ENGL_SENTENCE_SKILLS_SCORE	-0.0001179	0.0003011	0.695	=B5	95.58531	79.14261	67.67062	61.18592	59.4
ACCUPLACER_ENGL_READING_COMP_SCORE	0.00006708	0.0002833	0.813	=B6	88.68893	73.2381	61.08796	54.70374	51.35747
df=15918									
Multiple R-squared: 0.1009									
Success Prediction plugged with means					=$F4*G4 +$F5*G5 +$G6 +$F3	=$F4*H4+	=$F4*14+ $	=$F4*J4 +$	=$F4*K4+ $F5
prediction, x, from r (confirmation)					.4059203	.4043852	.4000365	.3979329	.4028048
x - se					=G10-G13	=H10-H13	=I10-I13	=J10-J13	=K10-K13
x + se					=G10+G13	=H10+H13	=I10+I13	=J10+J13	=K10+K13
se.fit					.0071281	.0086426	.0111202	.0127178	.01331346
Success rel to transfer level prediction					=G9/$G9	=H9/$G9	=I9/$G9	=J9/$G9	=K9/$G9
Count	=SUM (G15:K15)				7248	13174	11414	4827	882
Count of predicted successes with relative adjustment					=$B24* G15* G14	=$B24* H1	=B24*I15	=$B24* J15	=$B24* K15*
low est count of success					=G16	=$B24*H	=$B$24* I$	=$B$24* J$	=$B$24* K$15
high est count of success					=G17	=$B$24* H$	=$B$24* I$	=$B$24* J$	=$B$24* K$15
Adjusted success rate within level	=SUM(G16:K16)/SUM(G$15:K$15)				=G16/15	=H16/15	=I16/15	=J16/15	=K16/15
low est success rate	=SUM(G17:K17)/SUM(G$15:K$15)				=G17/15	=H17/15	=I17/15	=J17/15	=K16/15
high est success rate	=SUM(G18:K18)/SUM(G$15:K$15)				=G18/15	=H18/15	=I18/15	=J18/15	=K16/15
	Node Success	Regression	low error	high error					
Success rate in Statistics for HSGPA<2.3	0.43	=F19	=F20-F19	=F21-F19					

Appendix E: Adjusted success rate estimates by decision tree nodes for Transfer-Level English, Statistics, and Pre-calculus

While the focus of this paper is on the "lowest node", adjustments were performed for each node to aid overall success rate estimates. Note that in some representations of the MMAP findings, nodes were combined for improved clarity for a general audience and adjustments for combined nodes are shown along with each node adjustment.

Course	Node Level	Node description	N	Decision Tree Success Rate	Regression Adjusted Success Rate	Standard Error	Difference (Reg - Tree)	Percent of total N from regressions	Percent of total N in MMAP rules
Statistics	1	HS_11_GPA_CUM<2.3	49,046	40%	29%	3.7%	-11%	34%	12%
Statistics	2		17,267	49%	44%	1.5%	-5%	12%	10%
Statistics	3		32,541	58%	51%	1.1%	-7%	23%	12%
Statistics	4		4,013	70%	67%	0.9%	-3%	3%	4%
Statistics	5		40,386	80%	76%	2.4%	-4%	28%	62%
Statistics	2 & 3		49,808	55%	49%	1.2%	-6%	35%	22%
Statistics	2,3,4		53,821	56%	50%	1.2%	-6%	38%	26%
Statistics	4 & 5		44,399	79%	75%	2.3%	-4%	31%	66%
Pre-calculus	1		68,999	38%	28%	3.1%	-11%	48%	16%
Pre-calculus	2		3,281	49%	44%	1.8%	-5%	2%	5%
Pre-calculus	3		37,785	56%	50%	1.9%	-7%	26%	36%
Pre-calculus	4		16,710	67%	62%	1.5%	-5%	12%	21%
Pre-calculus	5		16,478	78%	75%	1.1%	-3%	12%	23%
Pre-calculus	2,3,4		57,776	59%	53%	1.8%	-6%	40%	62%
Transfer-level English	1		37,545	43%	42.6%	0.9%	0%	18%	10%
Transfer-level English	2		14,765	49%	47.5%	0.5%	-2%	7%	5%
Transfer-level English	3		57,054	61%	60.4%	0.6%	-1%	28%	23%
Transfer-level English	4		96,874	80%	78.6%	0.4%	-1%	47%	62%
Transfer-level English	2 & 3		71,819	59%	57.7%	0.6%	-1%	35%	28%

R^2 for Statistics = 0.10; R^2 for Pre-calculus = 0.09; R^2 for transfer-level English = 0.10

Research and Planning Group for California Community Colleges

The RP Group strengthens the ability of California community colleges to discover and undertake high-quality research, planning, and assessments that improve evidence-based decision-making, institutional effectiveness, and success for all students.

Technical Assistance

For technical assistance to replicate the analyses locally, questions regarding the code, or general questions, please contact:

Terrence Willett: twillett@rpgroup.org
Craig Hayward: chayward@rpgroup.org
Mallory Newell: newellmallory@deanza.edu
www.rpgroup.org