STRATEGIC DATA PROJECT

Understanding patterns of success among postsecondary CTE students: A diagnostic for institutional and system analysts

Coding Manual for R Users

This diagnostic was prepared with support from the ECMC Foundation. This diagnostic was prepared by Chris Avery, Jon Fullerton, Brian Johnson, Adrienne Murphy, Alyssa Reinhart, and Elise Swanson (listed alphabetically). The R translation and manual was prepared by Lauren Dahlin and Rachel Worsham.

Overview of CTE Toolkit Documentation

This coding manual for R users is intended as supplemental documentation for coding the analyses from the CTE Toolkit in R. Users of this guide should read the two primary sources of documentation for the CTE Toolkit prior to referring to this coding manual. The first source of documentation is the main narrative entitled "Understanding patterns of success among postsecondary CTE students: A diagnostic for institutional and system analysts." The narrative document introduces the Diagnostic: background, goals, terms, overviews of analyses, example visualizations, and suggestions for further reading. It is designed to be used with any statistical software, such as R, Stata, SPSS, etc., and does not contain any code. The second source of documentation is the "Technical Guide," which provides detail about how to execute the analyses.

Differences between the Technical Guide and Coding Manual for R Users

The visualizations that appear in the main narrative document and the technical guide were generated using Stata and a dataset from an education agency that is not available to the public. As a learning tool, a publicly-available simulated dataset is provided for R users. **The visualizations in the R coding manual are created using this simulated dataset and therefore have different underlying numbers from the technical guide and what appears in the main narrative**. For the most part, other than the differences between the datasets and the particulars of coding the analyses in R, the R version closely follows the Stata code. The R code included alongside this manual replicates the visualizations shown in the Narrative and Technical Guide. For some analyses, we provide suggestions for other disaggregates or interactions to further explore the data. Users may notice that the Stata files include code for these additional visualizations. We do not include code for these additional charts in the R code, as they usually require users to simply change variable names and graph labels to re-create these analyses. In this document, we identify places where analysts can tweak the R code to explore data further. Analysts will need to refer to the technical guide and R coding manual to complete analyses.

R is a free, open-source statistical software—meaning most of the functionality is created by R users. Analysts using R must download packages (also known as "libraries") for manipulating data, conducting statistical analysis, and creating visualizations. Although Stata-users can also download user-written programs, most analyses and data manipulation can be implemented without any user-written packages. In practice, this means that there are many alternatives to how an analysis can be coded in R, relative to Stata. For example, the visualizations in this technical guide are created using the R package called `ggplot2`; however, there are several graphical packages that can create similar visualizations, such as `plotly`. At OpenSDP, we try to use packages that are well-documented and commonly-used in the R-user community.

Setting Up Your R Environment

R can be used from the command line of a computer alongside any basic text editor. However, most analysts choose to use R with RStudio, which provides an integrated environment for writing and executing R code. When you download and install RStudio, you will also be prompted to download the latest version of R. This coding manual was created using RStudio version 2023.09.0+463 and base R version 4.3.1, but newer versions should continue to work, depending on whether newer versions of user-written packages remain backwards compatible.

At the beginning of the R code for each section, there is code to install and load any required packages. The list of packages is contained in the vector list called **`packages`**. If you do not already have the packages installed, you must uncomment the second line shown below to install the packages by removing the hash symbol (#) in front of it. After the packages are installed, it is good practice to re-insert the hash symbol so that R does not attempt to re-install the packages every time the analyst runs their R code.

```
packages<-c("tidyverse", "ggplot2", "nnet")

lapply(packages, install.packages, character.only = TRUE)
```

Note that the function **`lapply`** simply applies the function **`install.packages`** to each package in the list packages. The code could also be written as, for example, **`install.packages("tidyverse")`**, and repeated for each package.

Data and Key Considerations

The code in the R coding manual uses a simulated dataset based on a real agency. This dataset attempts to preserve relationships found in data from real students; for example, students who have higher high school GPAs tend to have better outcomes in CTE programs due to stronger prior preparation. Aside from these relationships, the data are randomly generated. Below, we provide an overview of the sample data and considerations for how to format and analyze data from your institution in a similar way.

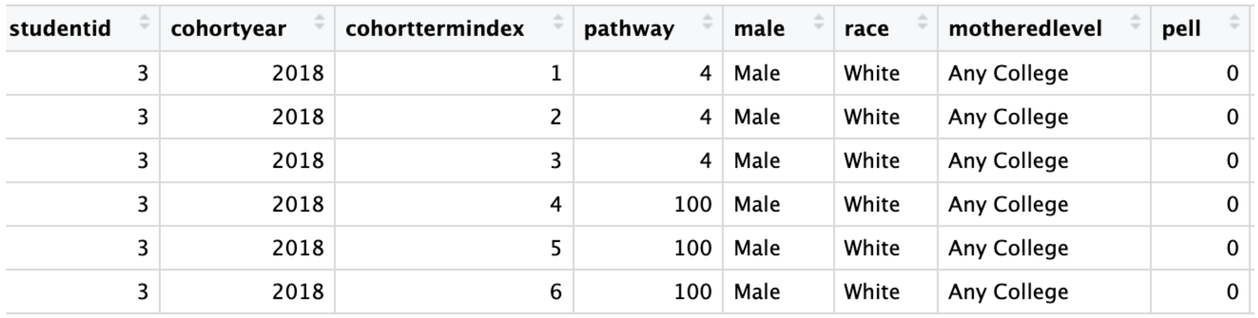
Note that due to the random generation of data, analyses that are adjusted for student characteristics using marginal effects or predicted probabilities are very similar to the same analyses that are not adjusted. Data that is not randomly generated will have larger differences between adjusted and unadjusted analyses.

The Sample Data

The sample dataset (found in the file **`data/faketucky_cte.csv`**) contains student characteristics (e.g. race/ethnicity and gender), student prior performance (e.g. high school GPA), CTE enrollment (e.g. the type of CTE program), and CTE outcomes of interest (e.g. degree completion or transfer). In R, data or matrices are stored in "data frames." We create the main data frame for the analyses **"`cte_data`"** by reading in the Faketucky CTE file using the code below.

```
cte_data <- read.csv(data_path)
```

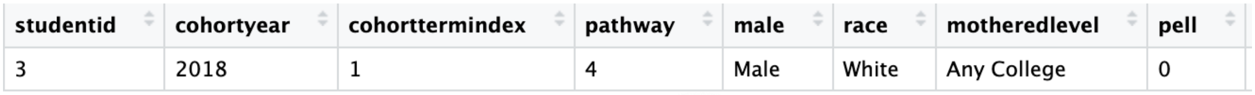
The dataset is unique for each student ID and term of enrollment. The data are manipulated such that every student appears in the data in each term, even after they graduate. After completion or transfer, the student has a pathway code of "100" (completion) or "101" (transfer). The student characteristics are stable across time—meaning they are repeated for each term. An example of this data is shown below for a single student. This student was enrolled in pathway four ("Mechanical Repair") and graduated after three terms.



studentid	cohortyear	cohorttermindex	pathway	male	race	motheredlevel	pell
3	2018	1	4	Male	White	Any College	0
3	2018	2	4	Male	White	Any College	0
3	2018	3	4	Male	White	Any College	0
3	2018	4	100	Male	White	Any College	0
3	2018	5	100	Male	White	Any College	0
3	2018	6	100	Male	White	Any College	0

For the analyses that do not look at student progression over time, we create a new data frame **`cte_data_stu`** that is unique at the student level using the code below. The code limits the data frame to the student's first term to make it unique at the student level using the **`filter`** function from **`tidyverse`**.

```
cte_data_stu <- cte_data %>%
  filter(cohorttermindex==1)
```



studentid	cohortyear	cohorttermindex	pathway	male	race	motheredlevel	pell
3	2018	1	4	Male	White	Any College	0

Refer to the main "Technical Guide" for key decision points in structuring the data from your institution. As you explore the analyses in R using the sample data, ask yourself:

- How is the data from my institution and population of interest, similar to or different from the sample data?
- What data manipulation is needed to put my data in a similar format to the sample data?

Data Manipulation Using R

Throughout this coding manual, we use the **`tidyverse`** package, which also contains the popular data-manipulation package **`dplyr`**, to create and manipulate data frames. The **`tidyverse`** package makes it much easier to create new columns and summarize data. Prior to beginning the analyses, we add some additional columns that are needed for our analysis. For example, consider the code below, which creates a new column with the entry pathway for each student (a value of "4" for the student example above):

```
cte_data <- cte_data %>%
  mutate(entry_pathway = ifelse(cohorttermindex==1, pathway, 0)) %>%
  group_by(studentid) %>%
  mutate(entry_pathway = max(entry_pathway)) %>%
ungroup()
```

First, using the mutate function, we create a new column that is the value of the students' pathway in the first term and zero otherwise. We want the entry pathway value to repeat for every student observation, so we **`group_by`** the student id to make the dataset one observation per student, take the maximum within **`studentid`**, and use **`ungroup`** to return the data to its original structure of one observation per student and term.

In addition to **`mutate`**, we often use **`summarize`** in a similar manner when operations are performed across multiple observations. For example, to create a column with the average of another column for each student, we would group the data using **`group_by(sid)`** and use the mean function inside summarize (e.g. **`summarize(something = mean(other)`**). The resulting dataset is unique for each student ID, but we could also use **`ungroup()`** to return the data to uniqueness on student ID and cohort term index. See the R code for examples.

Section 1: Data and Analysis Guide

Description of Analyses[1]

Are there differences in success rates across pathways?

In this analysis, we estimate the probability that a student in a given pathway will have a successful outcome event. We defined success in our analyses to be either completion of a credential of any form (associate degree, certificate, etc.) or transfer to another institution (depending on your context, you may want to specify this as transfer to a four-year institution or transfer to any institution).

The primary tool for Section 1 is a multinomial logistic regression model, a useful tool when there are multiple levels of the outcome variable. In this case, we have two levels of the outcome measuring student success: credential completion (1) or transfer (2). (These two levels are not inherently ordered; we could instead code transfer (1) and completion (2) without substantively altering the results.) The multinomial logistic regression allows us to estimate separate probabilities for each outcome in a single model, which is more efficient and allows us to account for the competing nature of the two types of success. Students are coded as either transferring or completing based on which event occurred first- for example, if a student completes a certificate and then transfers to a four-year institution, they are coded as a completer, not a transferer.

The first multinomial logistic regression model run expresses students' outcomes only as a function of their initial pathway choice. The multinomial logistic regression is implemented in R using the **`multinom`** function from the **`nnet`** package.

```
  model_pathway <- multinom(outcome ~ pathway, data = cte_data_stu, Hess = TRUE)
```

After the model is fit, the marginal effects (average predicted probabilities) for each entry pathway are computed using the **`predictions`** function from the **`marginaleffects`** package.

```
  model_pathway_margins <- predictions(model_pathway, by="pathway")
```

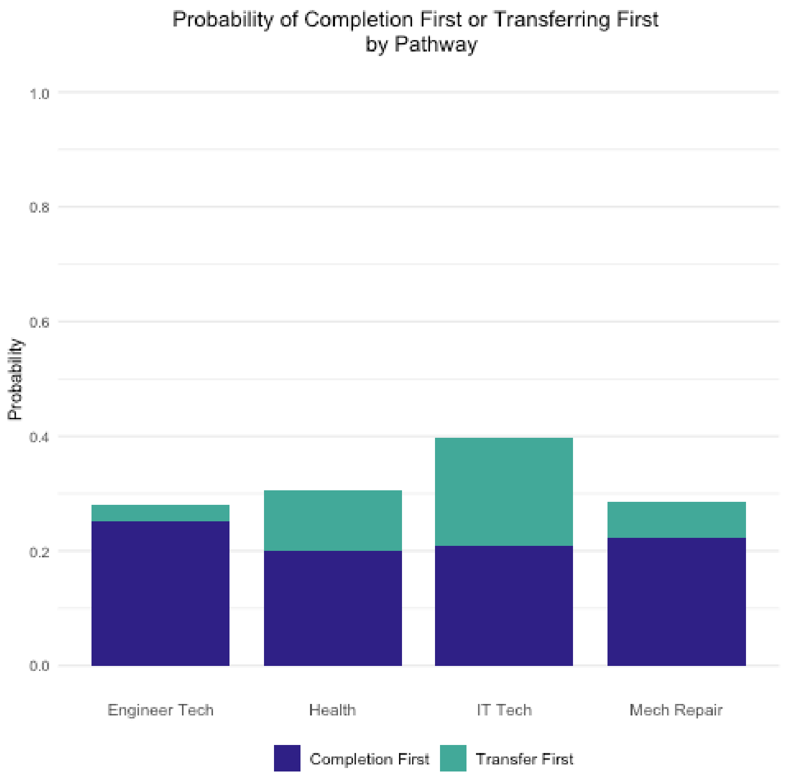
For the purpose of the visualization we then filter out the probability of not completing or transferring from the **`model_pathway_margins`** data frame. Finally, we plot the marginal effects using the **`ggplot`** package to create Example Visualization 1a.

```
  ggplot(model_pathway_margins, aes(fill = forcats::fct_rev(outcome), y=estimate,
  x=pathway)) +
    geom_bar(position="stack", stat="identity", width = 0.8) +
    scale_fill_manual("",
                      breaks = c("Completion First", "Transfer First"),
                      values = c(rgb(51, 34, 136, maxColorValue = 255),
                                 rgb(68, 170, 153, maxColorValue = 255))) + …con-
  tinued…
```

1 The descriptions of analyses for each section are abbreviated versions of the descriptions from the "Technical Guide." See this documentation for more details.

The code shown above creates a stacked bar chart using **`geom_bar`** with **`position="stack"`**. The **`stat`** is "**`identity`**" because we have already computed the statistic (i.e. **`ggplot`** does not need to compute a statistic, such as a mean, before creating the graph). The fill color is chosen based on the **`outcome`** variable and the colors are specified inside the **`scale_fill_manual`** option as **`rgb`** colors (encoding colors with values for red, green, and blue numbers). (Named colors, such as "red," instead of rgb values, can also be used.)

Example Visualization 1a



This figure shows whether students' likelihood of completion or transfer varies depending on what pathway they initially pursue. Critically, this is a descriptive analysis that does not account for student background characteristics, program context, or other factors that influence student outcomes. These results should not be interpreted to mean that one program is outperforming another, but rather to encourage deeper questioning and discussion across pathways.

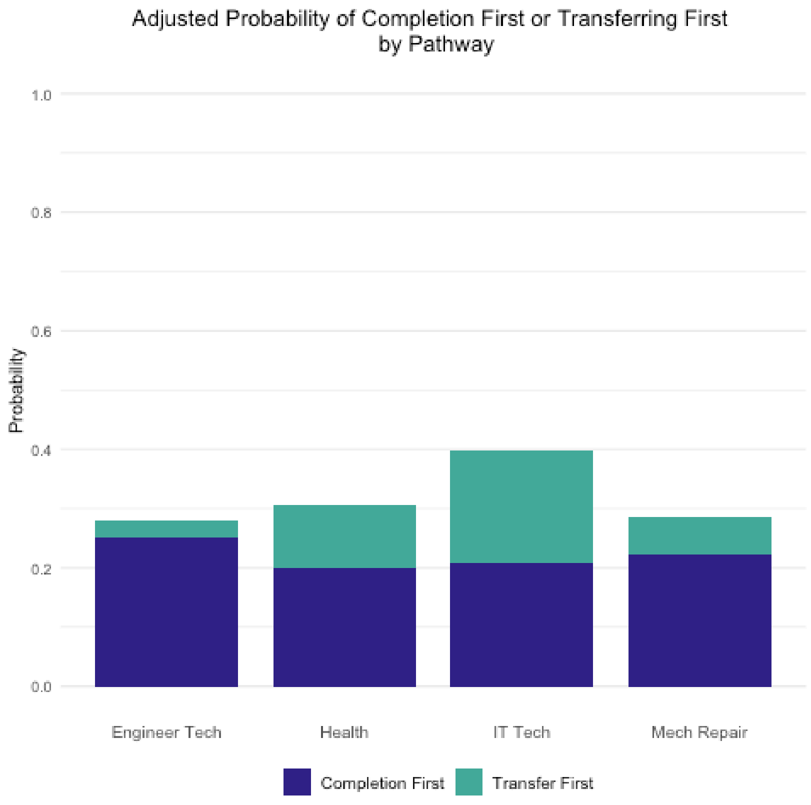
Are there differences for students with the same background characteristics?

The next stage of this analysis investigates potential differences in student success across pathways after accounting for students' background characteristics. Once again, we use a multinomial logistic regression model, with student outcome (success or transfer) as the dependent variable. The difference is that now we include several student-level variables, in addition to a student's initial pathway choice. These include gender (measured as a binary male/female), race/ethnicity (disaggregated to Asian, Black, Latina/o/x, white, and other/unknown), and high school GPA. Finally, we interact each of these additional control variables with a student's initial pathway choice, so that the relationships between student background characteristics and outcomes can vary by pathway. We run the model again using **`multinom`**.

```
  model_full <- multinom(outcome ~ hsgpa + mi_hsgpa + hsgpa*pathway +
                 male + male*pathway + motheredlevel +
    motheredlevel*pathway +
                  pell + mi_pell + pell*pathway +
                  race + race*pathway + pathway,
                  data = cte_data_stu, Hess = TRUE)
```

Like with the first multinomial logistic regression, we fit this model to student-level data, then predict probabilities of completion and transfer for each pathway, for a hypothetical student who has the average of each of the student traits included in the model. These results are again be averaged by pathway in a new dataframe. Finally, this dataset should be used to create another stacked bar chart with the adjusted probabilities of successful outcomes by pathway.

Example Visualization 1b



By design, this chart looks similar to Example Visualization 1a. As mentioned in the data section, Visualizations 1a and 1b look almost indistinguishable using our sample data because the data is randomly generated. Using real data, the useful information is in comparing how predicted probabilities of completion or transfer shift – or stay the same – after controlling, or "adjusting," for background student traits. If the predicted probabilities remain largely unchanged in a pathway, this is evidence that student factors in the model – race, gender, high school GPA, etc. – matter less in whether or not a student is successful. If none of the predicted probabilities shift very much across any of the pathways, this is evidence that differences in student success outcomes are potentially due less to which students select into particular pathways. Instead, differences in success could be due more to pathway-specific factors and structures.

If predicted probabilities within a pathway do differ after controlling for student traits, this is evidence that the selection of students into the pathway is explaining some of the observed success (or lack thereof).

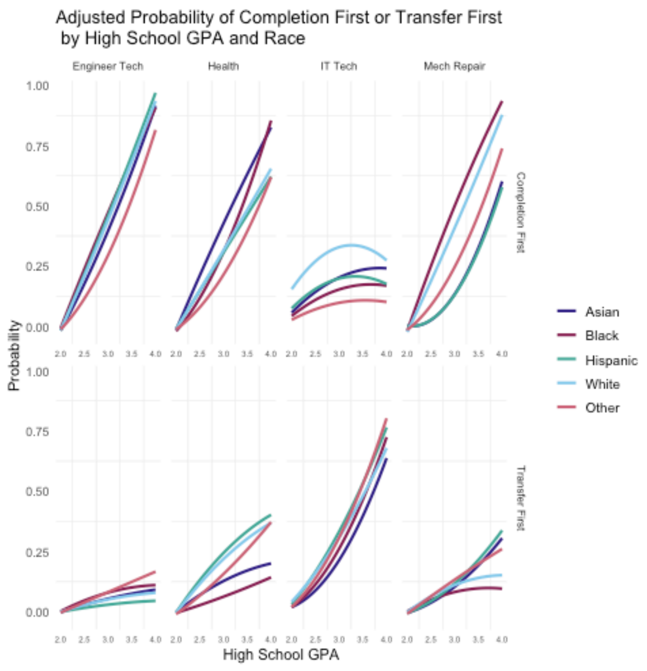
We can use the same regression model to further investigate, by pathway, which student factors are most associated with which success outcomes. Specifically, we can select one or two of the student traits included in the model, vary these while holding constant all the other traits, and see how predicted probabilities of success change. Visualization 1c uses the regression model to predict the probabilities of completion and transfer within each pathway, for different racial/ethnic groups, with varying high school GPAs, while holding fixed all other traits in the model. We again use the **`predictions`** function, specifying that we want the margins by **`pathway`**, **`race`**, and **`hsgpa`**.

```
model_full_hsgpa_margins <- predictions(
  model_full,
  newdata = datagrid(pathway = unique, race = unique, hsgpa = seq(2, 4, 0.25)))
```

To create the visualization, we use **`ggplot`** with additional functionality from another package, **`ggh4x`**. From **`ggplot`**, we create smoothed line graphs for the estimates by high school GPA with **`geom_smooth`**. The function **`facet_grid2`** from **`ggh4x`** is used to separate the plots by outcome and pathway and combine axis labels.

```
ggplot(model_full_hsgpa_margins, aes(hsgpa, estimate, group=race, color=race))
+
  geom_smooth(method = "lm", formula = y ~ poly(x, 2), se = FALSE) +
  scale_color_manual(values=group.colors) +
  facet_grid2(vars(outcome), vars(pathway), axes = "all", remove_labels = "y")
+ ...continued...
```

Example Visualization 1c

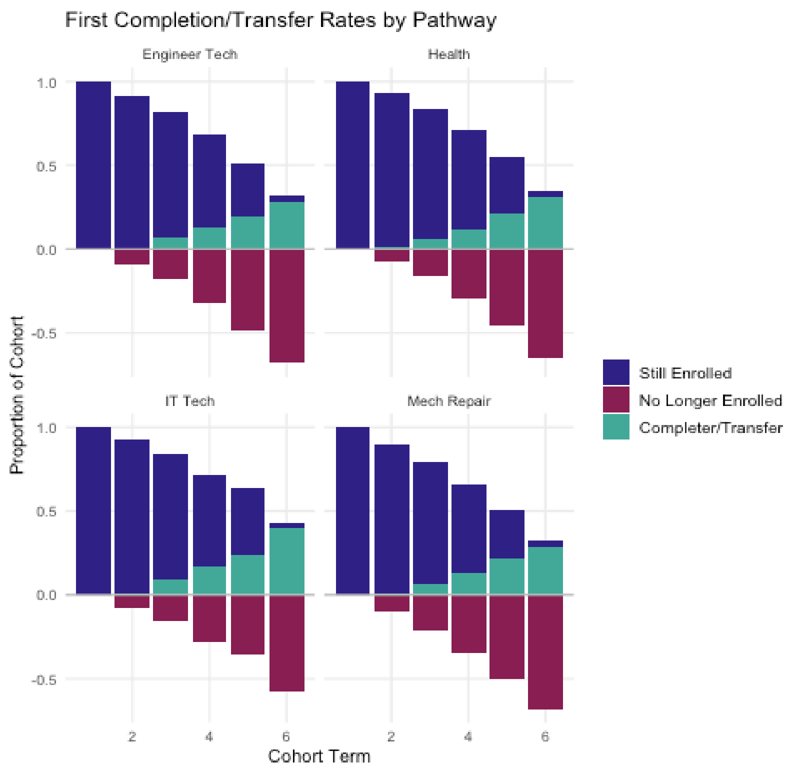


There is much we can learn from such a chart. One conclusion is that across pathway and racial/ethnic groups, a better high school GPA is less predictive of transfer than completion. In other words, across pathways and racial/ethnic groups, holding all other traits fixed, imagine two hypothetical students, one with a very good high school GPA and one with a more typical high school GPA. The student with the higher GPA has a much higher predicted probability of completing a credential than the lower-GPA peer, but a more modest predicted advantage in the probability of transfer compared to the lower-GPA peer. The STATA code and technical guide suggest other variables to try out in this chart, such as gender and GPA or Pell Grant status and gender. The code provided in the R manual allows for these tweaks and will require the analyst to change variable names and types in the margins calculation and plot.

Are there differences in the timing of success across pathways?

Visualization 1d displays the timing of student success, allowing you to compare not just the extent to which students are successful across pathways but the speed with which they achieve a successful outcome. The chart represents each cohort, in each pathway, in each term, as a single bar. The area of the bar in red below the x axis represents the proportion of students no longer enrolled. Above the x axis, the area of the bar represents the proportion of students still enrolled or those who have completed/transferred, with blue and turquoise distinguishing between the two. As the term is incremented, the graph conveys shifts in the proportion of students in each of the possible states (still enrolled, completer/transfer, no longer enrolled).

Example Visualization 1d



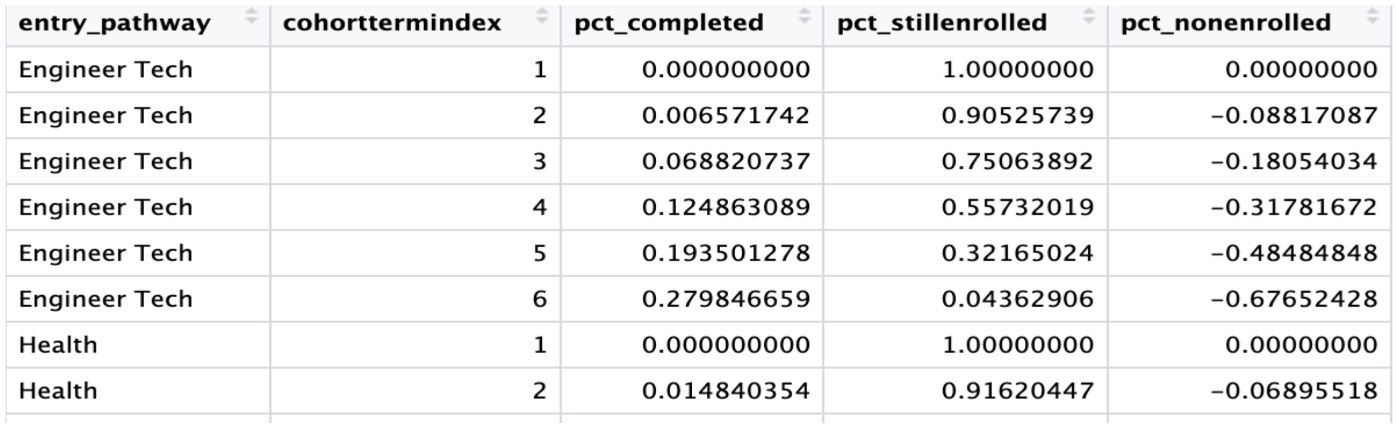
To create this chart in R, we must first create a new data frame with the average rates of non-enrollment, competition, and transfer by pathway and cohort term. The first dataframe created with **`tidyverse`** is "wide," showing the average rates for non-enrollment, completion, and transfer in each column. However, for the stacked bar chart using **`ggplot`**, it is easiest to graph the data in "long" format. We use the **`pivot_longer`** function from **`tidyverse`** to accomplish this transformation, shown below.

```
  waterfall_long <- waterfall %>%

    select(entry_pathway, cohorttermindex, pct_completed, pct_stillenrolled, pct_
  nonenrolled) %>%
    pivot_longer(
      cols = pct_completed:pct_nonenrolled,
      names_to = “status”,
      values_to = “value”
    ) %>%
    mutate(status = ifelse(status==”pct_stillenrolled”,”Still En-
  rolled”,ifelse(status==”pct_nonenrolled”, “No Longer Enrolled”,”Completer/
  Transfer”)))
```

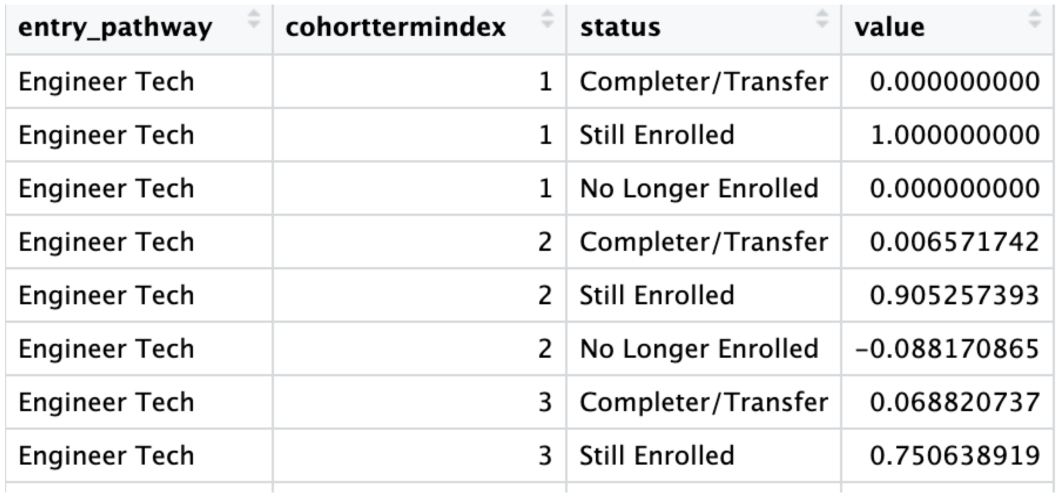
Transforming the waterfall data frame from "wide" to "long"

a. "Wide"



entry_pathway	cohorttermindex	pct_completed	pct_stillenrolled	pct_nonenrolled
Engineer Tech	1	0.000000000	1.00000000	0.00000000
Engineer Tech	2	0.006571742	0.90525739	-0.08817087
Engineer Tech	3	0.068820737	0.75063892	-0.18054034
Engineer Tech	4	0.124863089	0.55732019	-0.31781672
Engineer Tech	5	0.193501278	0.32165024	-0.48484848
Engineer Tech	6	0.279846659	0.04362906	-0.67652428
Health	1	0.000000000	1.00000000	0.00000000
Health	2	0.014840354	0.91620447	-0.06895518

b. "Long"



entry_pathway	cohorttermindex	status	value
Engineer Tech	1	Completer/Transfer	0.000000000
Engineer Tech	1	Still Enrolled	1.000000000
Engineer Tech	1	No Longer Enrolled	0.000000000
Engineer Tech	2	Completer/Transfer	0.006571742
Engineer Tech	2	Still Enrolled	0.905257393
Engineer Tech	2	No Longer Enrolled	-0.088170865
Engineer Tech	3	Completer/Transfer	0.068820737
Engineer Tech	3	Still Enrolled	0.750638919

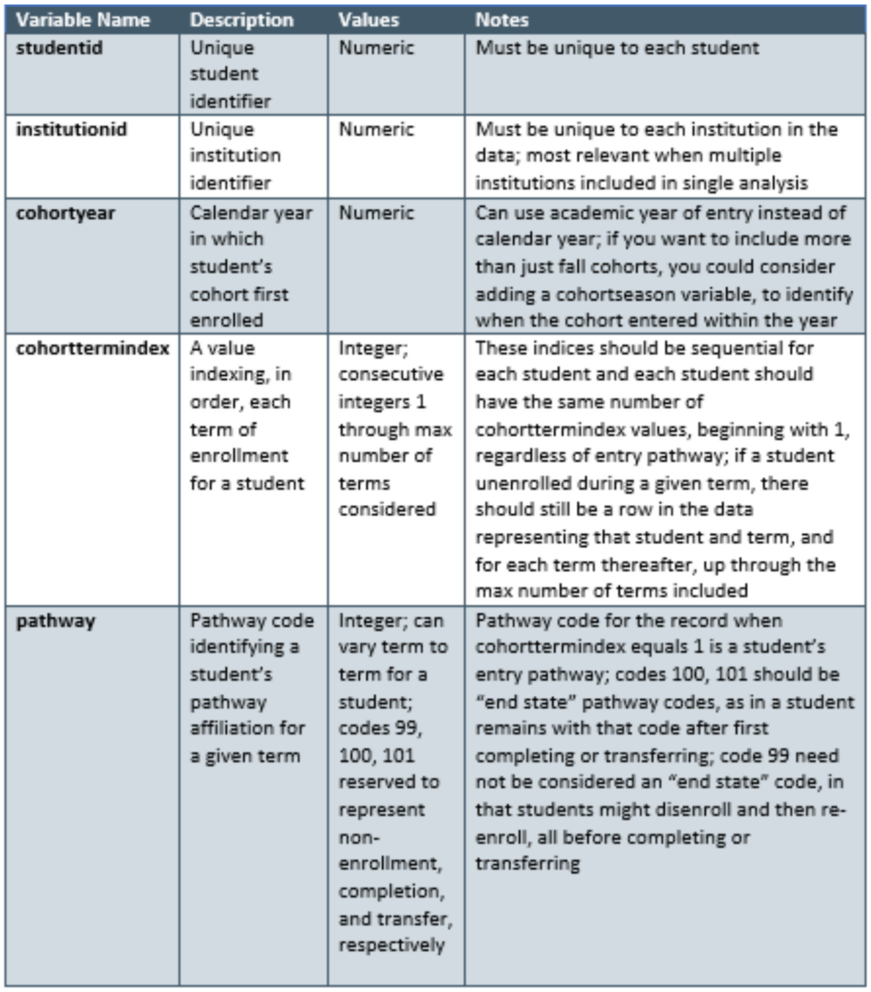
Level of Uniqueness

This file used in section 1 should be unique at the studentid-cohorttermindex level as described previously. In other words, each row in the data file should represent a unique term of enrollment for a unique student.

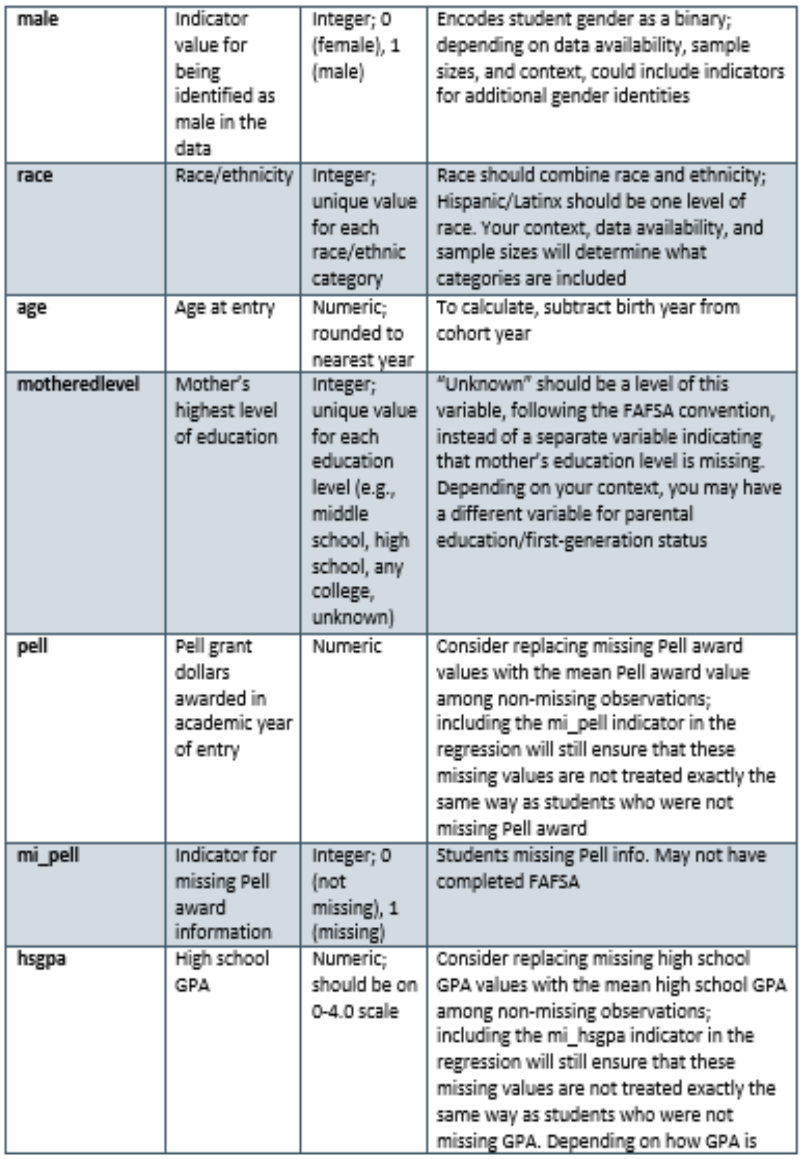
Only `cohorttermindex` and `pathway` values should vary over time for students in the data set. All other variables should be calculated based on the traits of a student at first entry into your institution and then held fixed for the rest of the terms, however many terms you decide to include.

We need multiple terms of data for each student in order to create the "waterfall" chart (see Example Visualization 1d), which requires observing a student over time. However, the multinomial logistic regression models only require a single observation per student (e.g., the first term, the term our sample code uses) because all the variables used in the model should be constant for a student over time: an outcome measured after however many terms you decide to include in your analysis, an initial pathway choice, and other covariates fixed at entry. In other words, you should not fit the regression models for Section 1 using all the observations in the data set, because each student will have multiple rows. See the R code for more details.

Data File Specification



Variable Name	Description	Values	Notes
studentid	Unique student identifier	Numeric	Must be unique to each student
institutionid	Unique institution identifier	Numeric	Must be unique to each institution in the data; most relevant when multiple institutions included in single analysis
cohortyear	Calendar year in which student's cohort first enrolled	Numeric	Can use academic year of entry instead of calendar year; if you want to include more than just fall cohorts, you could consider adding a cohortseason variable, to identify when the cohort entered within the year
cohorttermindex	A value indexing, in order, each term of enrollment for a student	Integer; consecutive integers 1 through max number of terms considered	These indices should be sequential for each student and each student should have the same number of cohorttermindex values, beginning with 1, regardless of entry pathway; if a student unenrolled during a given term, there should still be a row in the data representing that student and term, and for each term thereafter, up through the max number of terms included
pathway	Pathway code identifying a student's pathway affiliation for a given term	Integer; can vary term to term for a student; codes 99, 100, 101 reserved to represent non-enrollment, completion, and transfer, respectively	Pathway code for the record when cohorttermindex equals 1 is a student's entry pathway; codes 100, 101 should be "end state" pathway codes, as in a student remains with that code after first completing or transferring; code 99 need not be considered an "end state" code, in that students might disenroll and then re-enroll, all before completing or transferring



male	Indicator value for being identified as male in the data	Integer; 0 (female), 1 (male)	Encodes student gender as a binary; depending on data availability, sample sizes, and context, could include indicators for additional gender identities
race	Race/ethnicity	Integer; unique value for each race/ethnic category	Race should combine race and ethnicity; Hispanic/Latinx should be one level of race. Your context, data availability, and sample sizes will determine what categories are included
age	Age at entry	Numeric; rounded to nearest year	To calculate, subtract birth year from cohort year
motheredlevel	Mother's highest level of education	Integer; unique value for each education level (e.g., middle school, high school, any college, unknown)	"Unknown" should be a level of this variable, following the FAFSA convention, instead of a separate variable indicating that mother's education level is missing. Depending on your context, you may have a different variable for parental education/first-generation status
pell	Pell grant dollars awarded in academic year of entry	Numeric	Consider replacing missing Pell award values with the mean Pell award value among non-missing observations; including the mi_pell indicator in the regression will still ensure that these missing values are not treated exactly the same way as students who were not missing Pell award
mi_pell	Indicator for missing Pell award information	Integer; 0 (not missing), 1 (missing)	Students missing Pell info. May not have completed FAFSA
hsgpa	High school GPA	Numeric; should be on 0-4.0 scale	Consider replacing missing high school GPA values with the mean high school GPA among non-missing observations; including the mi_hsgpa indicator in the regression will still ensure that these missing values are not treated exactly the same way as students who were not missing GPA. Depending on how GPA is

Description of Analyses

When and why do students transfer across pathways and what are the outcomes of transfer?

The analysis in section 2 produces an interactive Sankey diagram showing trajectories of students across semesters. We start with a student-term level dataset in which students are grouped according to their initial pathway choice, in their entry term. Then, for each student and term, we get their status in the next term, which includes moving into, remaining in, or transitioning out of each possible pathway or outcome (unenrolled, still enrolled in first pathway, transferred to another pathway, transferred to another institution, or completed a credential). In R, we use the **`mutate`** and **`lead`** functions to get each student's status in the next term.

```
cte_data <- cte_data %>%
   mutate(pathway_target = ifelse(studentid==lead(studentid),
lead(pathway), NA))
```

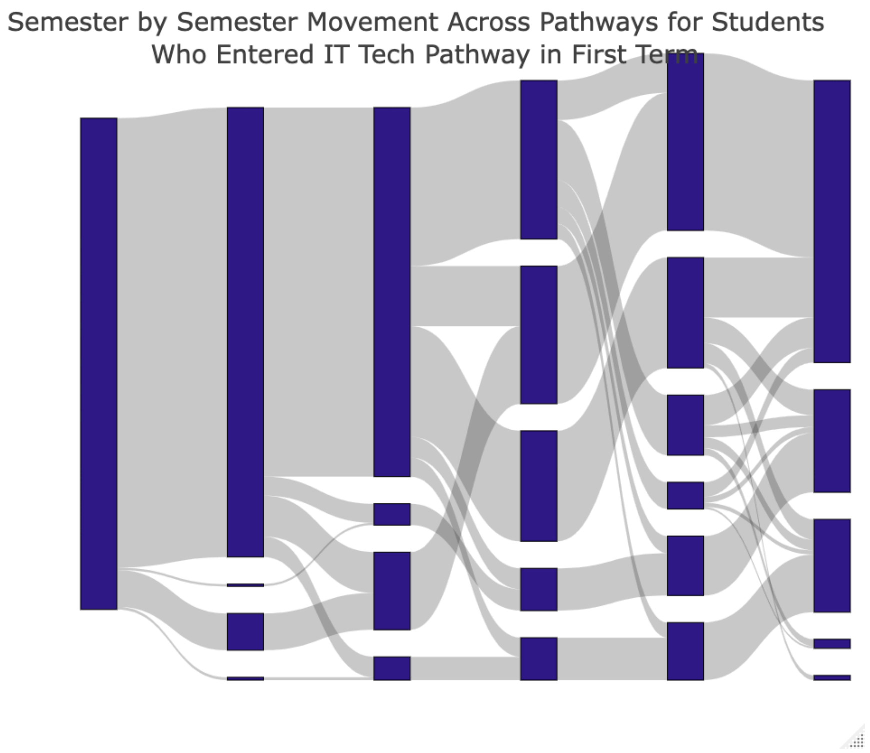
This dataset is then collapsed to the pathway-transition level to include aggregate counts of the number of students with each status in that term and the following term. The number of students by term, status, and status in the following term is calculated using the **`summarise`** function. Figure 2 on page 15 of the Technical Guide illustrates this transformation.

```
sankey_data <- cte_data %>%
  group_by(cohorttermindex, pathway, pathway_target) %>%
  summarise(num_students = n()) %>%
…
```

Next, we create a Sankey diagram to show the flow of students through pathways in each term. The diagram displays the size of each flow within each term transition, scaled to the number of students out of the starting total that the flow represents. We use the **`plotly`** package in R to create the Sankey diagram. Note that this is the same package that is used to create the Sankey diagram in the Stata version. The **`plotly`** package creates multiple types of charts, so we specify that the plot type is "sankey" and pass in a list of nodes and links.

```
fig <- plot_ly(
  type = "sankey",
  orientation = "h",
  arrangement="snap",
  node = list(
    label = nodes$label,
    … ),
  ),
  link = list(
 source = sankey_data$pathway_origin,
    target = sankey_data$pathway_target,
    value =  sankey_data$num_students
  )
)
```

Visualization 2a.

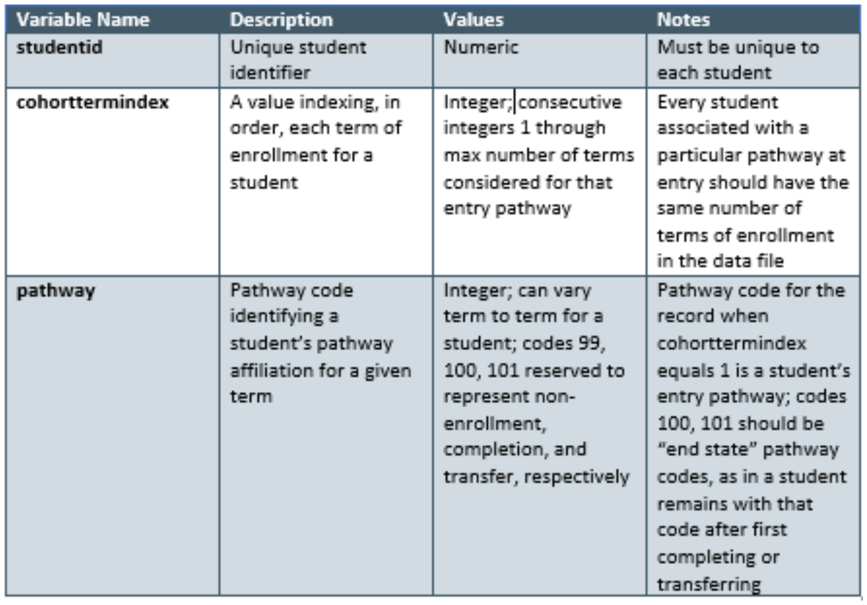


Using the **`plotly`** package enables interactivity such that you can hover your cursor over any particular flow and see the exact number of students making that transition between pathways/outcomes. See the **`plotly`** documentation for how to share interactive graphics.

Level of Uniqueness

This file should be unique at the studentid-cohorttermindex level. In other words, each row in the data file should represent a unique combination of **`studentid`** and **`cohorttermindex`**.

Data File Specification



Variable Name	Description	Values	Notes
studentid	Unique student identifier	Numeric	Must be unique to each student
cohorttermindex	A value indexing, in order, each term of enrollment for a student	Integer; consecutive integers 1 through max number of terms considered for that entry pathway	Every student associated with a particular pathway at entry should have the same number of terms of enrollment in the data file
pathway	Pathway code identifying a student's pathway affiliation for a given term	Integer; can vary term to term for a student; codes 99, 100, 101 reserved to represent non-enrollment, completion, and transfer, respectively	Pathway code for the record when cohorttermindex equals 1 is a student's entry pathway; codes 100, 101 should be "end state" pathway codes, as in a student remains with that code after first completing or transferring

Section 3: Data and Analysis Guide

Description of Analyses

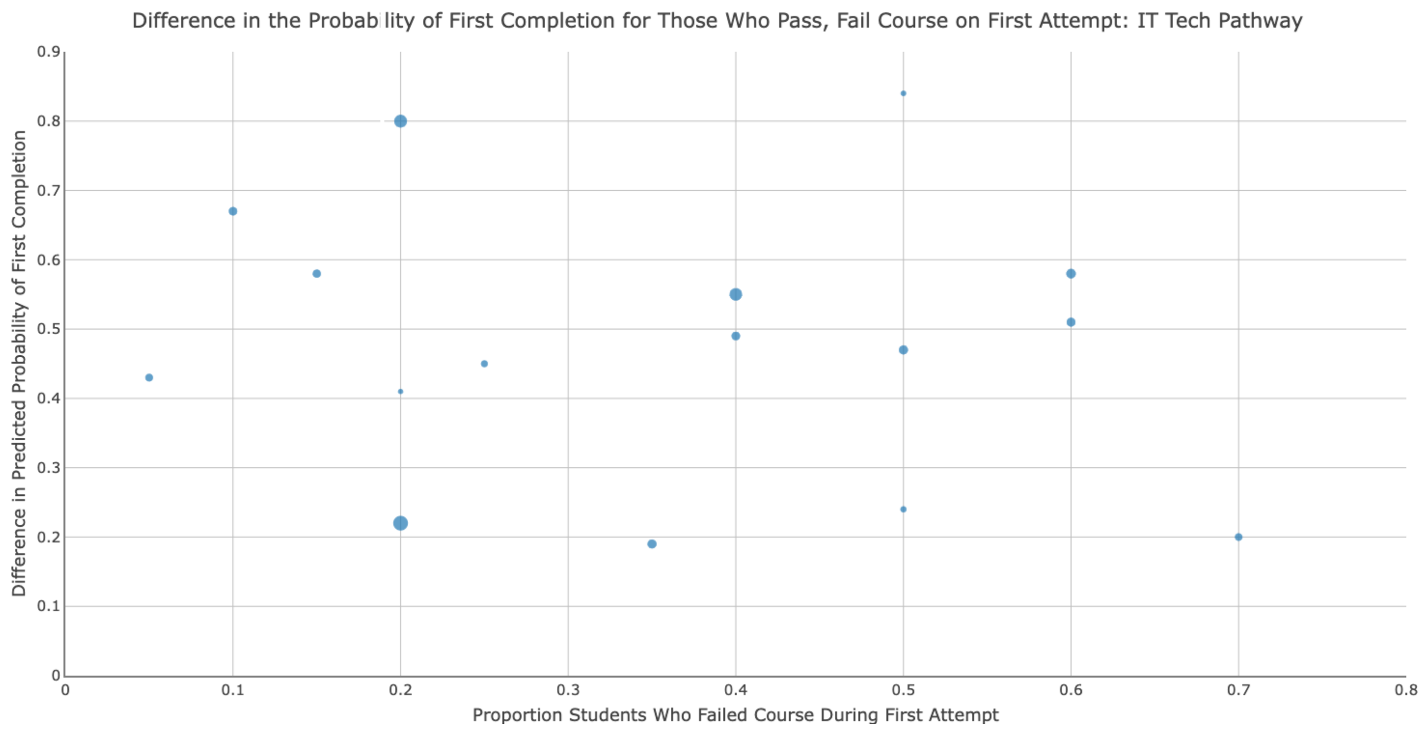
Are there gateway courses that are obstructing completion in some pathways?

This analysis investigates core courses that are required for students to complete within a pathway. These may be general requirements, such as introductory math or English, or pathway-specific requirements.

To replicate this analysis in R, we provide an example file that is unique for each course. See the Technical Guide pages 18 and 19 for an explanation of how to prepare the data from a student-course-year-level dataset. For each course, the data file shows the number of students attempting and succeeding in the required courses within their first three years of enrollment. Additionally, the file shows the share of students who completed a credential, among the students who passed and failed.

We then create an interactive scatterplot that plots the share of students failing a course against the change in probability of completing a credential if the course is failed initially. The size of each point reflects the number of students attempting the course. When interpreting this chart, we can think of large points with a high failure rate and a large decrease in probability of completion (e.g., a large point in the top right of the graph) as the most concerning from a student success perspective.

Visualization 3a



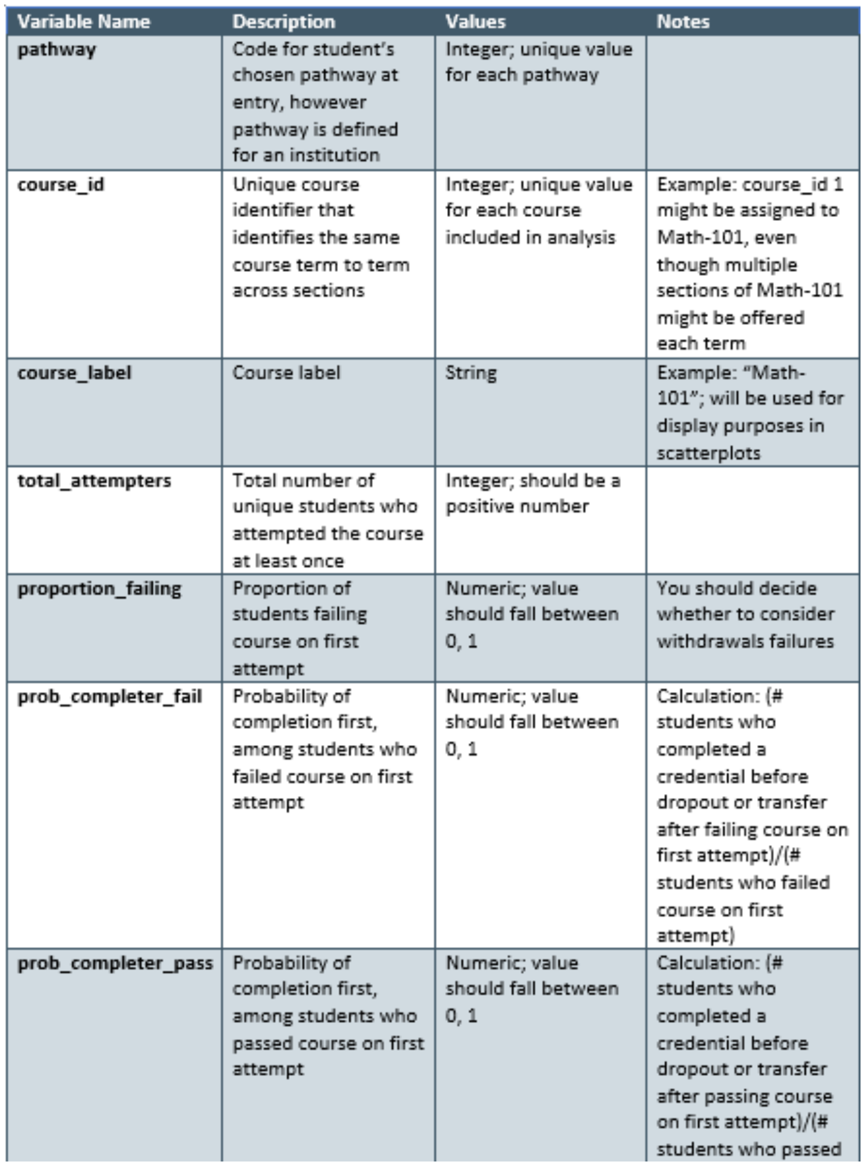
As with the Sankey diagram in Section 2, we use the **`plotly`** package to make the chart interactive. You can scroll over an individual point to see the course title, number of attempters, the probability of completion among those who initially pass, and the probability of completion among those who initially fail. To create the chart, we specify in the **`plotly`** function that the graph type is "scatter." We also create a custom hover over in the "text" argument.

```
fig <- plot_ly(cte_course_data,
               x = ~proportion_failing,
               y = ~prob_completer_diff,
               size = ~total_attempters,
               type = "scatter",
               mode = "markers",
          text = ~paste("Course Name: ", course_label,
                    "<br>Total Attempters: ", total_attempters,
                    "<br>Pass Probability: ", prob_completer_pass,
                    "<br>Fail Probability: ", prob_completer_fail),
               hoverinfo = "text"
)
```

Level of Uniqueness

This file should be unique at the pathway-course_id level. In other words, each row in the data file should represent a unique combination of pathway and course.

Data File Specification



Variable Name	Description	Values	Notes
pathway	Code for student's chosen pathway at entry, however pathway is defined for an institution	Integer; unique value for each pathway	
course_id	Unique course identifier that identifies the same course term to term across sections	Integer; unique value for each course included in analysis	Example: course_id 1 might be assigned to Math-101, even though multiple sections of Math-101 might be offered each term
course_label	Course label	String	Example: "Math-101"; will be used for display purposes in scatterplots
total_attempters	Total number of unique students who attempted the course at least once	Integer; should be a positive number	
proportion_failing	Proportion of students failing course on first attempt	Numeric; value should fall between 0, 1	You should decide whether to consider withdrawals failures
prob_completer_fail	Probability of completion first, among students who failed course on first attempt	Numeric; value should fall between 0, 1	Calculation: (# students who completed a credential before dropout or transfer after failing course on first attempt)/(# students who failed course on first attempt)
prob_completer_pass	Probability of completion first, among students who passed course on first attempt	Numeric; value should fall between 0, 1	Calculation: (# students who completed a credential before dropout or transfer after passing course on first attempt)/(# students who passed

Section 4: Data and Analysis Guide

Description of Analyses

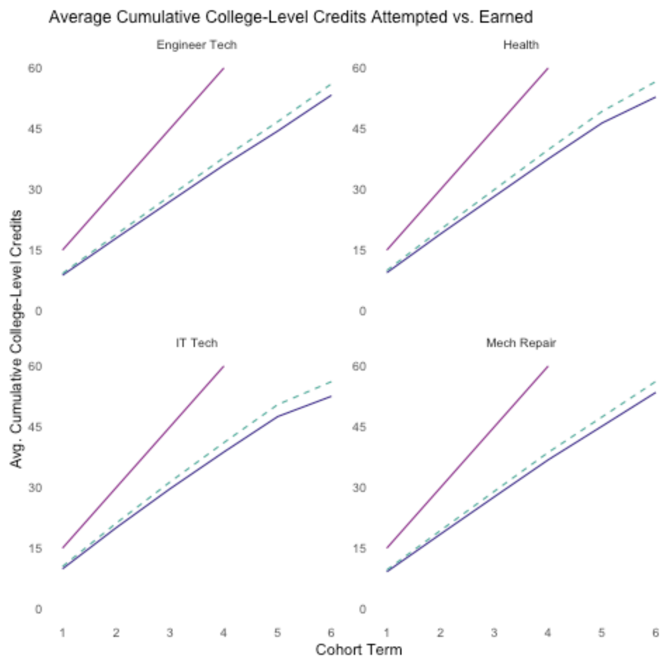
Why do pathway completion rates differ? How do students accumulate credits and progress in their pathway over time?

We begin this analysis by comparing college-level credit accumulation over time across pathways. We start with a student-term-course-level dataset and calculate, for each term, each student's total number of college-level (excluding developmental) credits earned and attempted. In R, we use **`mutate`** with the **`cumsum`** function to calculate the different types of cumulative credits for each student. For example, the code for cumulative college credits is shown below.

```
cte_credit_data <- cte_credit_data %>%
  group_by(studentid) %>%
  mutate(cumcollegecreditsearned =
ifelse(!is.na(collegecreditsearned),cumsum(collegecreditsearned),
NA))
```

After calculating the cumulative credits for each student, we then calculate each pathway's average cumulative college-level credits earned and attempted for each term, sorted with term ascending. The average cumulative credits for each term and pathway are stored in a new data frame, which we create using **`summarise`**, grouping by the term and pathway. Figure 4 on page 22 of the Technical Guide demonstrates this transformation. We are then ready to create visualization 4a using **`ggplot2`**.

Visualization 4a



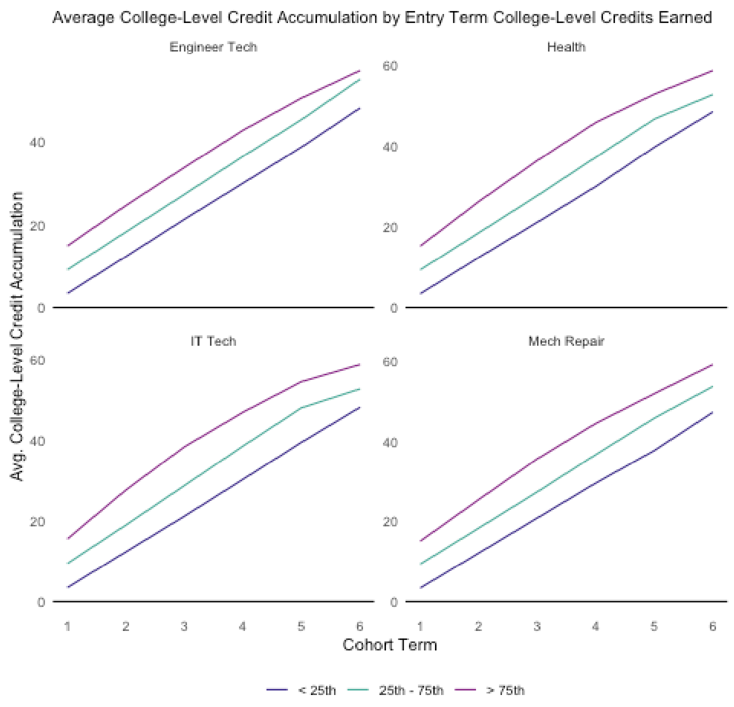
Note in the code for Visualization 4a, the use of **`facet_wrap(~ entry_pathway, scales = "free_y")`** to create a separate plot for each entry pathway.

In the next chart, we investigate the importance of early momentum in predicting continued progress. Specifically, we want a chart that plots, by pathway, average college-level credit accumulation among three groups of students: 1) those who were below the 25th percentile in the entry term for college-level credits earned, among other students entering the pathway 2) those who were between the 25th and 75th percentiles and 3) those who were above the 75th percentile. In R, we create these quartile bins by using the **`ntile`** function, then combining bins 2 and 3 (the 35th-75th percentiles).

```
first_term_cred$bin <-
ntile(first_term_cred$cumcollegecreditsearned,4)
```

where **`first_term_cred`** is a new dataframe of the first term only. This dataframe is then joined back onto the full dataframe for each student, so we know each student's bin of credits earned in the first term across all terms. As with Visualization 4a, the data is then collapsed by cohort term and pathway using **`summarise`**.

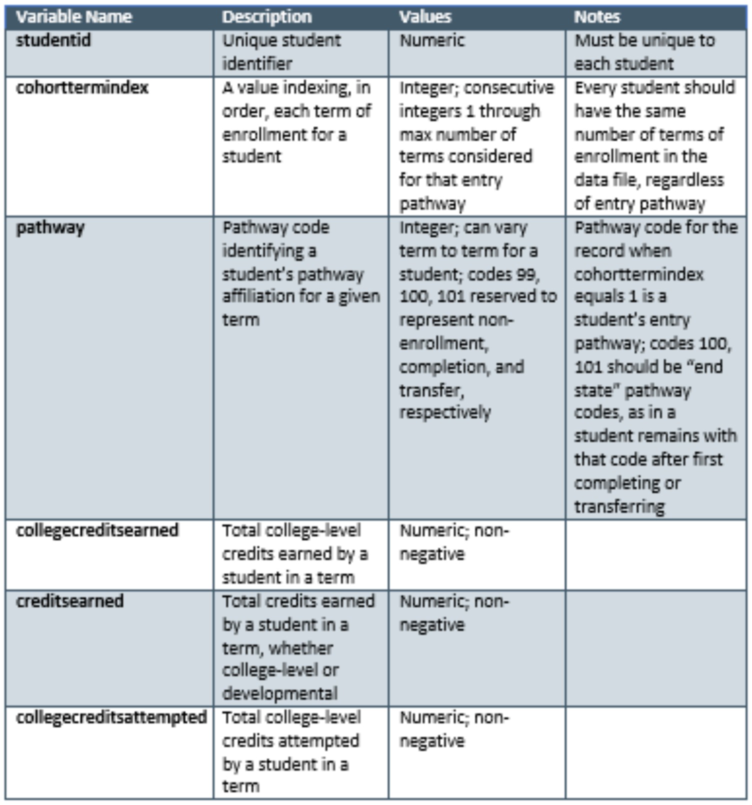
Visualization 4b



Level of Uniqueness

This file should be unique at the studentid-cohorttermindex level. In other words, each row in the data file should represent a unique combination of studentid and cohorttermindex.

Data File Specification



Variable Name	Description	Values	Notes
studentid	Unique student identifier	Numeric	Must be unique to each student
cohorttermindex	A value indexing, in order, each term of enrollment for a student	Integer; consecutive integers 1 through max number of terms considered for that entry pathway	Every student should have the same number of terms of enrollment in the data file, regardless of entry pathway
pathway	Pathway code identifying a student's pathway affiliation for a given term	Integer; can vary term to term for a student; codes 99, 100, 101 reserved to represent non-enrollment, completion, and transfer, respectively	Pathway code for the record when cohorttermindex equals 1 is a student's entry pathway; codes 100, 101 should be "end state" pathway codes, as in a student remains with that code after first completing or transferring
collegecreditsearned	Total college-level credits earned by a student in a term	Numeric; non-negative	
creditsearned	Total credits earned by a student in a term, whether college-level or developmental	Numeric; non-negative	
collegecreditsattempted	Total college-level credits attempted by a student in a term	Numeric; non-negative	

This concludes the R Coding manual.