College Pathway Analytics

INFO 5200 Learning Analytics Homework

[[Kimberly Williamosn, khw44]]

In this homework, you will learn how to analyze enrollment record data to identify patterns that can inform policy decisions about an academic curriculum and what information to provide to students as they plan their courses. You are given a synthetic dataset with an authentic correlation structure for students who have graduated in one of three majors (major 1, 2, and 3).

Learning Objectives

1. Understand how course enrollment data is structured
2. Identify hard course pairings using enrollment data
3. Identify course-major relationships to give students feedback about path-dependencies

Scenario

You are approached by a university administrator to provide input on upcoming policies about curriculum changes and information to students. You are asked to provide guidance on two high-level questions:

(1) Which courses should we advise students not to take in the same semester because it will be too difficult?

(2) What can we tell students about how their first-year course choices influence their likely major?

Data

The synthetic dataset contains one record per student course enrollment.

Variable	Data Type	Definition
student_id	numeric	Unique student identifier
major_id	numeric	Unique major identifier
course_id	numeric	Unique course identifier
term	numeric	Semester number in temporal order; e.g. 1=Fall 2017, 2=Spring 2018, 3=Fall 2018, etc.

Part 1. Understand the unique characteristics of course enrollment data

Question 1: How many (a) unique students, (b) unique courses, and (c) unique semesters are represented in the dataset?

```
#######################################
####### BEGIN INPUT: Question 1 #######
#######################################

# (a) unique students
length(unique(a$student_id))
```

```
## [1] 468
```

```
# (b) unique courses
length(unique(a$course_id))
```

```
## [1] 1490
```

```
# (c) unique semesters
length(unique(a$term))
```

```
## [1] 17
```

```
########################################
########################################
```

Question 2: What are the five most popular courses and how many students enrolled in each of them? How many courses have more than 20 enrollments overall; what proportion of all courses does this subset represent?

```
########################################
####### BEGIN INPUT: Question 2 #######
########################################

# Five most popular
a %>% group_by(course_id) %>%
  summarise(count=n()) %>%
  arrange(-count) %>% head(5)
```

```
## # A tibble: 5 x 2
##   course_id count
##       <dbl> <int>
## 1       185   361
## 2       186   354
## 3       193   324
## 4       980   307
## 5       192   297
```

```
# Number and proportion of courses with >20 enrollments
classesOver20 = a %>% group_by(course_id) %>%
  summarise(count=n()) %>%
  filter(count>20) %>%
  nrow()

classesOver20
```

```
## [1] 112
```

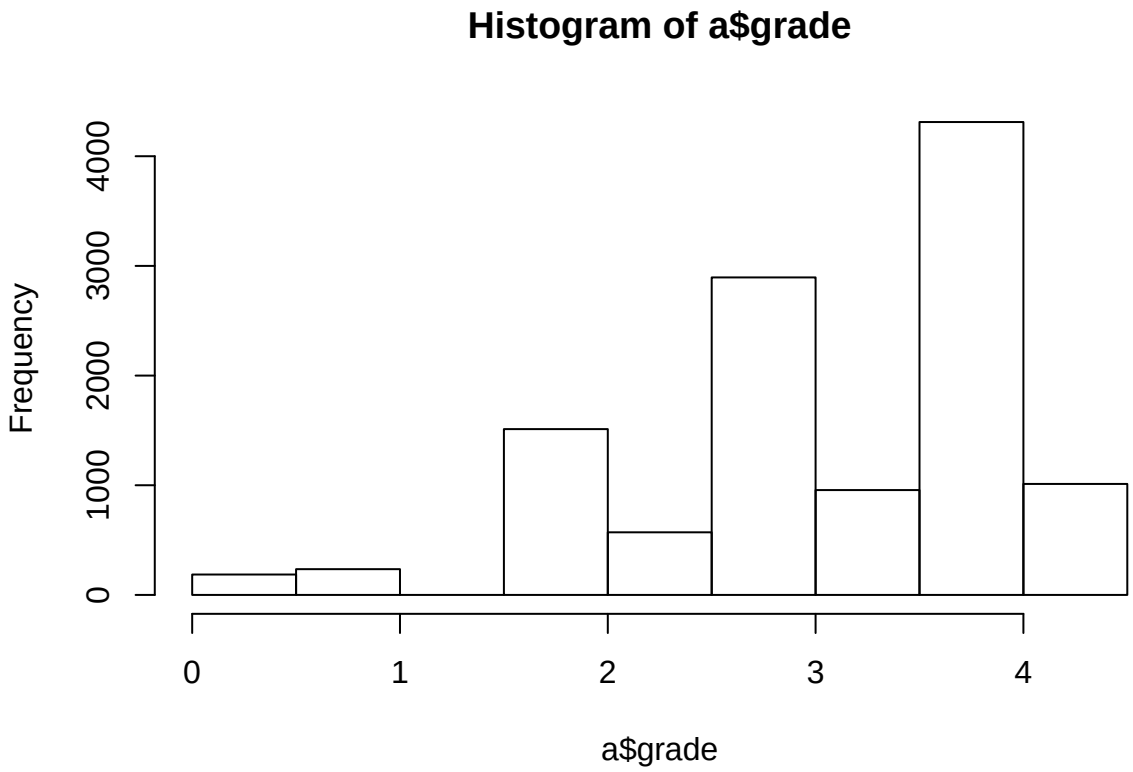
```
classesOver20/(a %>% group_by(course_id) %>%
  summarise(count=n()) %>%
  nrow())
```

```
## [1] 0.07516779
```

```
########################################
########################################
```

Question 3: Plot the overall grade distribution as a histogram. What is the overall average grade?

```
########################################
####### BEGIN INPUT: Question 3 #######
########################################

# Distribution
hist(a$grade)
```




```
# Average grade
mean(a$grade)
```

```
## [1] 3.205083
```

```
#####################################
#####################################
```

Part 2. Which course combinations are associated with lower grades when taken in the same semester?

The goal is to identify course pairings that should be avoided because students have earned lower grades when taking them together compared to taking them some time apart.

Before moving forward, it makes sense to narrow the set of observations to courses that are frequently chosen (say at least 20 times) and terms where a student took more than just one course. I do this for you in the code below. Work with the `tcp` dataframe for the rest of Part 2.

```
tcp = a %>%
    group_by(course_id) %>% filter(n() > 20) %>%
    group_by(student_id, term) %>% filter(n() > 1)
```

Question 4: For each student, for each semester, identify all of their course pairings (i.e. if a student took courses A, B, and C in a semester then the pairings are AB, BC, and AC). Use the `tcp` dataset. You can use two `for` nested loops over students and semesters to achieve this. The `combn(x, m=2)` function returns all possible pairings for a vector `x`. To keep adding new rows of data to an existing dataframe as you loop, you can use `pairs = rbind(pairs, my.new.rows)`. For all the pairings you identify, keep track of the student ID and semester. You should get 7724 course pairs in total.

```
pairs = data.frame()
```

```
#######################################
####### BEGIN INPUT: Question 4 #######
#######################################
# add your code here
students = unique(tcp$student_id)
for(student in students){
  terms = tcp %>% subset(student_id==student, select=c(term)) %>% unique() %>% pull()
  for(termt in terms){
    courseTermSelection = tcp %>% filter(student_id==student & term==termt)
    courseCombinations = combn(courseTermSelection$course_id, m=2, simplify = FALSE)
    newData = data.frame(student_idP=student, termP=termt,
                         t(data.frame(courseCombinations)), row.names = NULL)
    pairs = rbind(pairs, newData)
  }
}
#######################################
#######################################

head(pairs)
```

```
##   student_idP termP   X1   X2
## 1         124    13  661  663
## 2         124    14  412 1197
## 3         124    14  412   61
## 4         124    14 1197   61
## 5          73    15  186  185
## 6          73    15  186  750
```

Question 5: For each pair of courses you found above, compute the average grade of the two courses that the student received. Use a `for` loop to go over each pair, get the grade out of `tcp` for both courses for that student in that semester, and then average the two grades. You can save it in a new column in the `pairs` dataset. The average of all the average grades should be 3.088. Note that in this dataset there are cases where the same course appears multiple times for the same student in the same term but with different grades (this is an artifact of noisy real-world data); you should simply average over all of the grades for a given student/term/pair of courses.

```
pairs$avg_grade = NA

#######################################
####### BEGIN INPUT: Question 5 #######
#######################################
# add your code here
for(pair in 1:nrow(pairs)){
  grade1 = tcp %>% filter(student_id==pairs[pair,"student_idP"] & term==pairs[pair,"termP"] &
                          course_id==pairs[pair,"X1"]) %>% ungroup() %>% pull() %>% mean()
  grade2 = tcp %>% filter(student_id==pairs[pair,"student_idP"] & term==pairs[pair,"termP"] &
                          course_id==pairs[pair,"X2"]) %>% ungroup() %>% pull() %>% mean()
  pairs[pair,"avg_grade"] = mean(c(grade1, grade2))
}
#######################################
#######################################

head(pairs)
```

```
##   student_idP termP   X1   X2 avg_grade
```

```
## 1         124    13  661  663     3.000
## 2         124    14  412 1197     3.835
## 3         124    14  412   61     3.835
## 4         124    14 1197   61     3.670
## 5          73    15  186  185     2.000
## 6          73    15  186  750     2.000
```

```
mean(pairs$avg_grade)
```

```
## [1] 3.085999
```

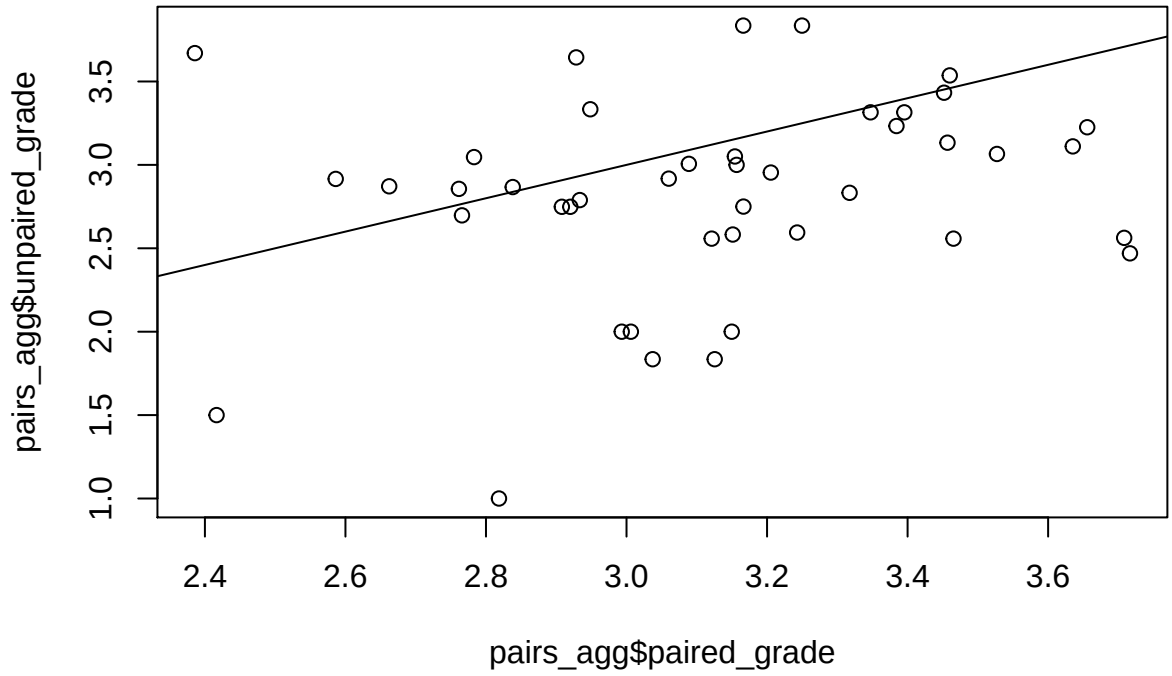
Question 6: Now aggregate your `pairs` dataset to have 1 row = 1 course pair, each with the average grade across all students/terms and the frequency of pair occurrence. You should use `group_by()` and `summarise()` for this computation. Ignore cases where it looks like a student took the same course twice in one semester. Remove all course pairings that have come up less than 20 times. You should find 43 such course pairs. Be sure to count pairs independent of their order, i.e. 14 and 19 is the same as 19 and 14.

```
#######################################
####### BEGIN INPUT: Question 6 #######
#######################################
for(pair in 1:nrow(pairs)){
  course1 = pairs[pair,"X1"]
  course2 = pairs[pair,"X2"]
  if(course1>course2){
    pairs[pair,"X1"] = course2
    pairs[pair,"X2"] = course1
  }
}

pairs_agg = pairs %>% group_by(X1,X2) %>%
  summarise(count=n(), paired_grade=mean(avg_grade)) %>%
  filter(count>19 &X1!=X2)


#######################################
#######################################
```

Question 7: For each of the 43 course pairs (i.e. `for` loop), find students in the entire `a` dataframe who took both of those courses (i.e. `course_id %in% c(first, second)`) but not in the same semester (i.e. `n_distinct(term) == 2`). Compute the average grade across both courses each student received and then average those average grades for each of the 43 course pairs. Report which FOUR pairs of courses students should avoid taking in the same term the most because they have much lower average grades when taken in the same semester.

```
pairs_agg$unpaired_grade = NA

#######################################
####### BEGIN INPUT: Question 7 #######
#######################################

# add your code here
for(pair in 1:nrow(pairs_agg)){
  course1 = pairs[pair,"X1"]
  course2 = pairs[pair,"X2"]

  studentsWBothCourses = intersect(a %>% filter(course_id==course1) %>% pull(student_id),
            a%>%filter(course_id==course2) %>% pull(student_id))
```

```
  studentsWBothCoursesInSameTerm = intersect(a %>% filter(course_id==course1) %>%
                                               select(c(student_id, term)),
            a%>%filter(course_id==course2) %>% select(c(student_id, term))) %>%
    pull(student_id)

  studentsWBothCoursesDiffTerm=intersect(studentsWBothCourses, studentsWBothCoursesInSameTerm)

   meanGrades = vector()
   for(stu in studentsWBothCoursesDiffTerm){
    meanGrade = a %>% filter(student_id==stu & course_id %in% (c(course1,course2))) %>%
      pull() %>% mean()
    meanGrades = append(meanGrades,meanGrade)
   }
  pairs_agg[pair,"unpaired_grade"] = mean(meanGrades)
}

# Compare the paired and unpaired average grade for each course pair.
plot(pairs_agg$paired_grade, pairs_agg$unpaired_grade); abline(0,1)
```




```
# Co-enrollment pairs to avoid:

# add your code here
mutate(pairs_agg, diff=paired_grade-unpaired_grade) %>% arrange(diff)
```

```
## # A tibble: 43 x 6
## # Groups:   X1 [13]
##       X1    X2 count paired_grade unpaired_grade    diff
##    <dbl> <dbl> <int>        <dbl>          <dbl>   <dbl>
##  1     8   934    22         2.39           3.67 -1.28
##  2   193   934    28         2.93           3.64 -0.716
##  3     8   193    27         3.17           3.84 -0.669
##  4     8   192    22         3.25           3.84 -0.585
```

```
##  5   950   952    42         2.95           3.33 -0.385
##  6   585   934    56         2.59           2.92 -0.330
##  7   185   585    53         2.78           3.05 -0.264
##  8   585   946    35         2.66           2.87 -0.209
##  9   185   934    51         2.76           2.86 -0.0948
## 10   193  1126    25         3.46           3.54 -0.0764
## # ... with 33 more rows
```

```
# Four pairs to avoid:

# add them here
# 8 and 934
# 193 and 934
# 8 and 193
# 8 and 192
#######################################
#######################################
```

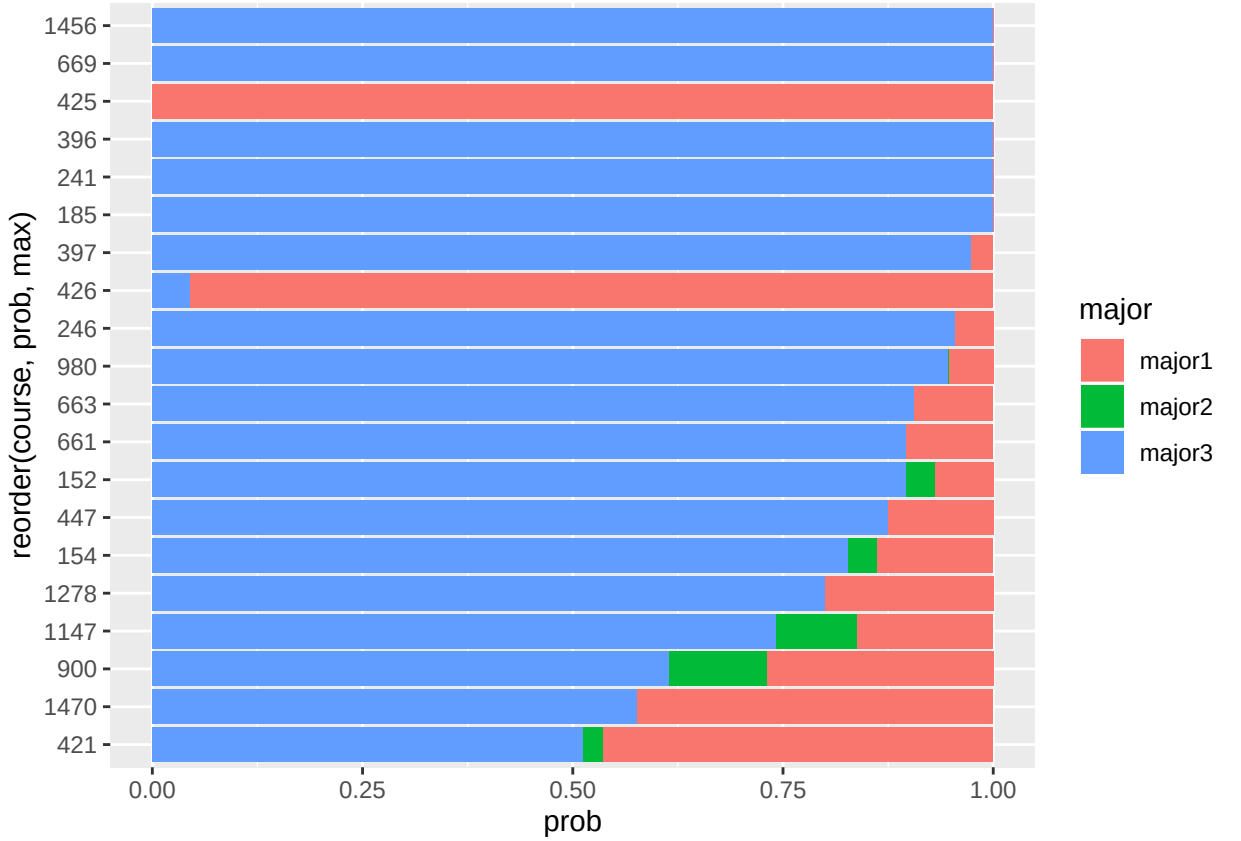
Part 2: How do students' first-year course choices influence their likely major?

Note: I am showing you how to do this, so follow the code carefully, and at the end **there is one result interpretation question**. Don't forget to answer it and write your self-reflection below.

For the courses that students commonly take in their first term, how does the choice of which ones they enroll in influence their likelihood of majoring in a field?

```
# First, identify the most commonly taken courses in the student's first term for all students.
# This is relative to the student, not simply term=1.
# Define 'commonly taken' as over 20 first-term enrollments.
comm = a %>%
    group_by(student_id) %>%
    filter(term == min(term)) %>%
    group_by(course_id) %>%
    count %>%
    filter(n > 20) %>%
    arrange(-n)

# Second, compute the likelihood that a student majors in each of the three majors
# conditional on enrolling in the first term in each one of the classes identified above.
# Thus, you are computing 3 * number of classes probabilities.
ftrm = a %>%
  group_by(student_id) %>%
  filter(term == min(term)) # keep each student's 1st term data

major = data.frame()

for(cr in comm$course_id) {
  tmp.major = ftrm %>%
            filter(course_id == cr) %>%
            group_by(student_id) %>%
            slice(1) %>%
            ungroup %>%
            summarise(
```

```
                course = cr,
                major1 = mean(major_id == 1),
                major2 = mean(major_id == 2),
                major3 = mean(major_id == 3)
            )

  major = rbind(major, tmp.major)
}

# Third, make a visualization that shows the likelihood of majoring in each major (1,2,3)
# after taking each of the identified courses in the first term. Try to make a bar plot
# with stacked bars for each course and color fill shows the major distribution.
major_lng = gather(major, major, prob, 2:4)
ggplot(major_lng, aes(reorder(course, prob, max), prob, fill = major)) +
    geom_bar(stat="identity") + coord_flip()
```



Question 8: Completing the statements below by interpreting the final plot above:

```
########################################
####### BEGIN INPUT: Question 8 #######
########################################

# - Students who take course 669 are most likely to major in major 3.
# - Students who take course 425 are most likely to major in major 1.
# - Students who take course 421 have about equal probability of majoring in major 1 and major 3.

########################################
```

```
#####################################
```

Self-reflection

Briefly summarize your experience on this homework. What was easy, what was hard, what did you learn?

I found the homework a little confusing. Specifically question 5 where we are not told how to handle duplicates yet.

Submit Homework

This is the end of the homework. Please **Knit a PDF report** that shows both the R code and R output and upload it on the EdX platform. Alternatively, you can Knit it as a "doc", open it in Word, and save that as a PDF.

Important: Be sure that all your code is visible. If the line is too long, it gets cut off. If that happens, organize your code on several lines.