MI 320 Reasoning With Data
Week 14 Lab: April 10, 2019

Instructions:

- Please read all of these instructions before you begin working.
- Create a new R Markdown file to save your work for this lab. To create a new R Markdown file, use the RStudio menus to create a new .Rmd file: *File > New File > R Markdown…*
- The header of your R Markdown file should look similar to the following:

```
---
title: "title that describes this document"
author: "write your name here"
date: "write today's date here"
output: html_document
---
```

- To receive full credit for this lab, you must submit **three** files to the appropriate location on D2L: your R Markdown file, the HTML file produced when you "Knit" your R Markdown file, and your R history file.
- If you do not submit all three files, what you turned in will be graded and then 50% of the total points you earned for the lab will be subtracted from your score.
- You must turn in your lab by 11:59pm on Saturday April 13 to receive full credit. If your lab is not turned in by the time it is due, you will receive 0 points for the lab.
- In your R Markdown file, each "part" of the lab should have its own level 2 heading, and each question should have its own level 3 heading.
- Each question should have its own code chunk with the code for that question in it.
- If a question asks for an answer instead of, or in addition to, writing R code, you should type your answer as text in the body of your R Markdown file, not inside a code chunk.
- Your HTML file that results from knitting your Rmd file should display both the R code for each question, and the output from executing the R code.
- If any of the code you turn in for the lab produces an error or otherwise does not run correctly, you can still receive partial credit if you include a comment indicating that you are aware of the error, and describing what you tried to do to correct the error and answer the question.

Part 1: Definitions

Write a definition for each term below. Use complete sentences.

1. Relational data

2. Primary key

3. Mutating join

4. Filtering join

Part 2: Describe the data

The dataset you will use for this lab comes from some speed dating experiments that were conducted by researchers at Columbia University, about people's dating preferences.

If you are unfamiliar with what speed dating is, see the Wikipedia page (https://en.wikipedia.org/wiki/Speed_dating) about it.

The data files and documentation for this lab are located on D2L:

- speed_dating_documentation.txt
- people.csv
- dates.csv

You will need to download these files, and save them in the same folder as this .Rmd file. Make sure you load BOTH of the data files in the setup chunk of your R Markdown file. Don’t forget to name the setup chunk, and make it so that the output of the R commands in the setup chunk does NOT display when you knit your file to HTML.

Relational datasets have more than one kind of observation in the data, and variables that can be used to connect multiple data tables together to answer questions.

5. What are the kinds of observations in the dataset?

6. What are the relations in the dataset?

7. What are the primary keys in each table of the speed dating dataset? Note that it is possible that one or both tables do not have a key, or that the primary key consists of more than one variable.

8. Verify using count() that the iid variable is a "primary key" in the people table. How can you tell that iid is a primary key?

9. Add a "surrogate key" to the dates table using mutate() and row_number(). Make sure you save the resulting data table back to the "dates" object.

Part 3: Left joins

10. Write R code to create a data table that contains all of the columns of the *dates* table, and all of the columns of the *people* table, matched up by the the *iid* variable. Store the result of your R command as a new object called "dates.people". Note: you should use a left join for this.

11. Using the new *dates.people* data table that you created, write R code that uses select() to select all the columns that originally came from the *dates* table, and just the race variable that originally came from the *people* table. Overwrite the *dates.people* object with the output of the select command.

12. Write R code to create a data table that contains all of the columns of the *dates.people* table you created, and just the *iid* and race variables from the *people* table, matched up by the the *pid* variable in the *dates.people* table and the *iid* variable in the *people* table. Overwrite the *dates.people* object with the output of this command.

    To do this you will need to do another left join like the one you did above, this time between the new *dates.people* object, and a subset of the columns in the *people* table, matching up by the *pid* variable in *dates.people* and the *iid* variable in *people*. The *pid* variable is the id of the partner in the speed dating, and the *pid* values in the *dating* table match up with the *iid* values in the *people* table.

    Note: I have started this command for you below. **You need to fill in the necessary information in the left_join() command.** Note that you must join by two variables that have different names.

    ```
    people.subset <- people %>% select(iid, race)
    dates.people <- dates.people %>%
      left_join() ## fill in the necessary information in the left_join
    command
    dates.people
    ```

13. Still using the *dates.people* object, write R code that uses select() to select the following columns that originally came from the *dates* table (*iid, pid, round, position, match, like*), and the two race columns. Overwrite the *dates.people* object with the output of the select() command.

Part 4: Answer a question using relational data

14. You now have a data table that you can use to answer the following question: Is being the same race as one's partner important for whether people "match" in speed dating? Write the necessary R code to manipulate the data table so that you can answer this question.

    To do this, write a dplyr command that uses mutate() to add a new logical column to the *dates.people* table containing TRUE if the participant and his or her partner are the same race (if the values in the two race columns in that row are equal), and FALSE if they are different (if the values in the two race columns in that row are NOT equal). Name this new variable "same.race".

15. Write a ggplot2 command to create a bar graph that uses the new "same.race" variable you just created in the *dates.people* table, and the "match" variable.

    The match variable contains the data about whether both partners said yes, they would like to see that partner again. The bar graph that you create below will show you how many people who were the same race as their partner had a "match", and how many did not.

    Note: Don't forget to group by the two variables, and use summarize to count the number of observations in each group. The "fill" should be the "same.race" variable.

16. Is being the same race as one's partner important for whether people "match" in speed dating? How can you tell?

Extra Credit (not required)

Students take multiple courses per semester. So in a real system that contains data about which courses people are taking, each person would be associated with more than one course at at time. This is called a one-to-many relationship; one course can be taken by many people at the same time.

Copy and paste the code on the next page into an R chunk in your R Markdown file. This code creates three tables that are similar to the ones created in the Rmd file for the lecture, except there's three tables instead of two:

- The *people* table contains just the information about the people registered for the courses.
- The *courses* table contains just the information about the courses, like the title, when they take place, the instructor, etc.
- The *people.courses* table contains the information about which students are registered for which courses.

```
people <- tribble(
  ~id, ~last.name, ~first.name,
  1, "Smith", "Alexander",
  2, "Williams", "Olivia",
  3, "Bell", "Emma",
  4, "Thompson", "Liam",
  5, "Garcia", "Sophia"
)

courses <- tribble(
  ~id, ~course.num, ~title, ~schedule, ~instructor, ~location,
~credits, ~prereqs,
  1, "MI220", "Understanding Users", "M W 3:00 PM - 4:20 PM", "Wash",
"233 Communication Arts Bldg", 3, NA,
  2, "MI250", "Intro Applied Programming", "M W 10:20 AM - 12:10 PM",
"Introne", "C134 Holden Hall", 3, NA,
  3, "MI320", "Reasoning with Data", "W 12:40 PM - 3:30 PM", "Rader",
"106 Farrall Ag Eng Hall", 3, "MI 220 and MI 250",
  4, "MI350", "Evaluating HCT", "Tu Th 10:20 AM - 12:10 PM", "Rader",
"233 Communication Arts Bldg", 3, "MI 220 and MI 250",
  5, "MI420", "Interactive Prototyping", "Tu Th 10:20 AM - 12:10 PM",
"Wyche", "C134 Holden Hall", 3, "MI 220 and MI 250"
)

people.courses <- tribble(
  ~person.id, ~course.id,
  1, 1,
  1, 2,
  2, 3,
  2, 4,
  3, 4,
  3, 5,
  4, 1,
  4, 2,
  5, 3,
  5, 4
)
```

After you have run the code to create the tables, a dplyr command that uses joins to create another data table that has more than one row per student, and the columns: *last.name, first.name, course.num, title, schedule, instructor, location, credits, prereqs*. This new data table lists which courses each student is registered for.

When you are finished, turn in three files on D2L:

- Your R Markdown file. Make sure your name is in your file.
- Your HTML file. To generate the HTML file, click “Knit” in the R Studio interface.
- Your R History file.