E401/M518: Problem Set 3a

Data Management

Fall 2023

Due: November 14 2023

Please work on the following questions and hand in your solutions in groups of at most 2 students. You are asked to answer all questions, but I will only select 2 questions randomly to grade.

Part 1: R questions

Question 1: Data import and tidying

Tibbles and data frames

1. How can you tell if an object is a tibble? (Hint: try printing `mtcars`, which is a regular data frame).

2. Compare and contrast the following operations on a `data.frame` and an equivalent tibble. What is different? Why might the default data frame behaviors cause you frustration?

```r
df <- data.frame(abc = 1, xyz = "a")
df$x
df[, "xyz"]
df[, c("abc", "xyz")]
```

Data import

1. What function would you use to read a file where fields were separated with "|"?

2. What are the most important arguments to `read_fwf()`?

3. Sometimes strings in a CSV file contain commas. To prevent them from causing problems they need to be surrounded by a quoting character, like `"` or `'`. By default, `read_csv()` assumes that the quoting character will be `"`. If you need to customize the import options, the general import command `read_delim()` might be the better choice. What arguments do you need to specify to read the following text into a data frame?

```
"x,y\n1,'a,b'"
```

Parsing vectors

1. What happens if you try and set `decimal_mark` and `grouping_mark` to the same character? What happens to the default value of `grouping_mark` when you set `decimal_mark` to `,`? What happens to the default value of `decimal_mark` when you set the `grouping_mark` to `.`?

1. What are the most common encodings used in Europe? What are the most common encodings used in Asia? What are the most common encodings in your home country? Do some googling to find out.

Spreading and gathering

1. Why are `gather()` and `spread()` not perfectly symmetrical? Carefully consider the following example (Hint: look at the variable types and think about column names.) :

```
stocks <- tibble(
  year   = c(2015, 2015, 2016, 2016),
  half  = c(   1,    2,     1,    2),
  return = c(1.88, 0.59, 0.92, 0.17)
)
stocks %>%
  spread(year, return) %>%
  gather("year", "return", `2015`:`2016`)
```

2. Both `spread()` and `gather()` have a `convert` argument. What does it do?

3. Why does this code fail?

```
table4a %>%
  gather(1999, 2000, key = "year", value = "cases")
```

4. Why does spreading this tibble fail? How could you add a new column to fix the problem?

```
people <- tribble(
  ~name,             ~key,    ~value,
  #-----------------|--------|------
  "Phillip Woods",   "age",       45,
  "Phillip Woods",   "height",   186,
  "Phillip Woods",   "age",       50,
  "Jessica Cordero", "age",       37,
  "Jessica Cordero", "height",   156
)
```

5. Tidy the simple tibble below. Do you need to spread or gather it? What are the variables?

```
preg <- tribble(
  ~pregnant, ~male, ~female,
  "yes",     NA,    10,
  "no",      20,    12
)
```

Separating and uniting

1. What do the `extra` and `fill` arguments do in `separate()`? Experiment with the various options for the following two toy data sets.

```
tibble(x = c("a,b,c", "d,e,f,g", "h,i,j")) %>%
  separate(x, c("one", "two", "three"))

tibble(x = c("a,b,c", "d,e", "f,g,i")) %>%
  separate(x, c("one", "two", "three"))
```

2. Both `unite()` and `separate()` have a `remove` argument. What does it do? Why would you set it to `FALSE`?

Missing values

1. Compare and contrast the `fill` arguments to `spread()` and `complete()`.
2. What does the direction argument to `fill()` do?

Question 2: Relational data and data types

The following questions use several tables from the `nycflights13` data discussed in class.

Relational data

1. Imagine you wanted to draw (approximately) the route each plane flies from its origin to its destination. What variables would you need? What tables would you need to combine?
2. In the diagram on the lecture slides, I forgot to draw the relationship between `weather` and `airports`. What is the relationship and how should it appear in the diagram?
3. `weather` only contains information for the origin (NYC) airports. If it contained weather records for all airports in the US, what additional relation would it define with `flights`?
4. We know that some days of the year are special, and fewer people than usual fly on them. How might you represent that data as a data frame? What would be the primary keys of that table? How would it connect to the existing tables?

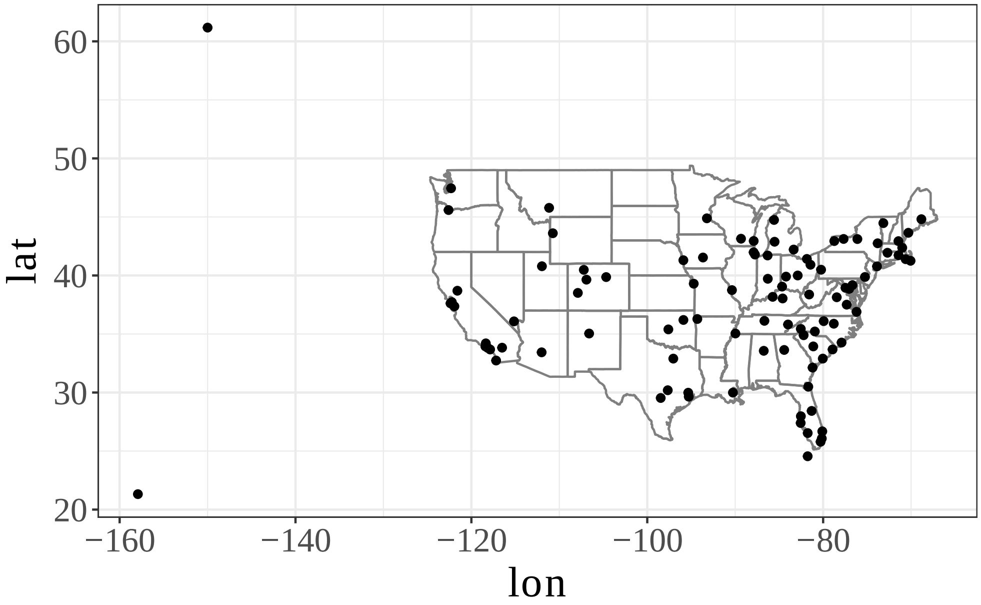
Keys

1. Add a surrogate key to `flights`.

Mutating joins

1. Compute the average delay by destination, then join on the `airports` data frame so you can show the spatial distribution of delays. Below is an easy way to draw a map of the United States (be sure to install and load the required `maps`-package first!):

```
airports %>%
  semi_join(flights, c("faa" = "dest")) %>%
  ggplot(aes(lon, lat)) +
    borders("state") +
    geom_point() +
    coord_quickmap()
```



   Don't worry if you don't understand the details of the above code. You might want to use the `size` or `colour` of the points to display the average delay for each airport.

2. Add the location of the origin *and* destination (i.e. the `lat` and `lon`) to `flights`.

3. Is there a relationship between the age of a plane and its delays?

4. What happened on June 13 2013? Display the spatial pattern of delays, and then use Google to cross-reference with the weather.

Filtering joins

1. Filter flights to only show flights with planes that have flown at least 100 flights.
2. Find the 48 hours (over the course of the whole year) that have the worst delays. Cross-reference it with the `weather` data. Can you see any patterns?
3. You might expect that there's an implicit relationship between plane and airline, because each plane is flown by a single airline. Confirm or reject this hypothesis using the tools you've learned above.

Strings

1. In code that doesn't use the `stringr`-package, you'll often see `paste()` and `paste0()`. What's the difference between the two functions? What `stringr`-function are they equivalent to? How do the functions differ in their handling of `NA`?
2. In your own words, describe the difference between the `sep` and `collapse` arguments to `str_c()`.
3. What does `str_wrap()` do? When might you want to use it?
4. What does `str_trim()` do? What's the opposite of `str_trim()`?

Part 2: Your project

Question 3

This question is basically handing in the proposal for your course project as discussed in the first lecture. I will definitely check this part and you should take it seriously since the project counts for a significant part of your course grade.

In at most two pages, summarize your research question, the data and empirical methods you will use, and any important preliminary findings you may have at this stage.

In addition, try to organize your data management in a systematic way and hand in your code. I would like to see the following:

- Systematic handling of the data read-in, i.e., your code should take the original data files as input.
- (If applicable) a systematic combination and organization of the distinct data sources that you may be using based on the relational database concepts that we discussed.
- Systematic cleaning: Have you checked that the data doesn't contain any errors, missing values, etc? How do you handle these in your code? Why did you choose this strategy? Have you ensured that all variables have the correct type?