Relational Data Assignment

Tyler Kephart

2024-08-15

```
library(tidyverse)
```

Required Libraries

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

Question 1. Identify the primary keys in the following datasets. Be sure to show that you have the primary key by showing there are no duplicate entries.

Lahman::Batting babynames::babynames nasaweather::atmos

Note: I had to `install.packages("nasaweather")` to get the dataset

```
# primary key would be a combination of the following columns
# or create a new column based on the row number to be the primary key
Lahman::Batting %>% count(playerID, yearID, stint, teamID) %>% filter(n > 1)
```

```
## [1] playerID yearID   stint    teamID   n
## <0 rows> (or 0-length row.names)
```

```
# again primary key would be a combo of the following or create from row numbers
babynames::babynames %>% count(year, sex, name, n) %>% filter(nn > 1)
```

```
## Storing counts in `nn`, as `n` already present in input
## i Use `name = "new_name"` to pick a new name.
```

```
## # A tibble: 0 x 5
## # i 5 variables: year <dbl>, sex <chr>, name <chr>, n <int>, nn <int>
```

```
# again primary key would be a combo of the following or create from row numbers
nasaweather::atmos %>% count(lat, long, year, month) %>% filter(n > 1)
```

```
## # A tibble: 0 x 5
## # i 5 variables: lat <dbl>, long <dbl>, year <int>, month <int>, n <int>
```

Question 2. What is the relationship between the "Batting", ~~"Master"~~ "People", and "Salaries" tables in the "Lahman" package? What are the keys for each dataset and how do they relate to each other?

```
# we already got the primary key combo for Lahman::Batting in question 1
# Lahman::Batting key combo is playerID, yearID, stint, and teamID
# Lahman::People primary key is playerID
Lahman::People %>% count(playerID) %>% filter(n > 1)
```

```
## [1] playerID n
## <0 rows> (or 0-length row.names)
```

```
# Lahman::Salaries key combo is yearID, teamID, and playerID
Lahman::Salaries %>% count(yearID, teamID, playerID) %>% filter(n > 1)
```

```
## [1] yearID   teamID   playerID n
## <0 rows> (or 0-length row.names)
```

```
# all 3 share playerID, Salaries and Batting also share yearID and teamID
```

Question 3. Load the "nycflights13" library. Use an appropriate join to add a column containing the airline name to the "flights" dataset present in the library. Be sure to put the carrier code and name in the first two columns of the result so we can see them. Save the result as "flights2".

```
# loading nycflights13
library(nycflights13)
# join airlines to flights to add airline name
flights2 <- flights %>%
            left_join(airlines, "carrier") %>%
            # put carrier code and airline name first to confirm
            select(carrier, airline_name = name, everything())
flights2
```

```
## # A tibble: 336,776 x 20
##    carrier airline_name       year month   day dep_time sched_dep_time dep_delay
##    <chr>   <chr>             <int> <int> <int>    <int>          <int>     <dbl>
##  1 UA      United Air Lines~  2013     1     1      517            515         2
##  2 UA      United Air Lines~  2013     1     1      533            529         4
##  3 AA      American Airline~  2013     1     1      542            540         2
##  4 B6      JetBlue Airways    2013     1     1      544            545        -1
##  5 DL      Delta Air Lines ~  2013     1     1      554            600        -6
##  6 UA      United Air Lines~  2013     1     1      554            558        -4
##  7 B6      JetBlue Airways    2013     1     1      555            600        -5
##  8 EV      ExpressJet Airli~  2013     1     1      557            600        -3
```

```
##  9 B6      JetBlue Airways    2013     1     1      557            600        -3
## 10 AA      American Airline~  2013     1     1      558            600        -2
## # i 336,766 more rows
## # i 12 more variables: arr_time <int>, sched_arr_time <int>, arr_delay <dbl>,
## #   flight <int>, tailnum <chr>, origin <chr>, dest <chr>, air_time <dbl>,
## #   distance <dbl>, hour <dbl>, minute <dbl>, time_hour <dttm>
```

Question 4. Use an appropriate join to add the airport name to the "flights2" dataset you got above. The codes and names of the airports are in the "airports" dataset of the "nycflights13" package. Put the carrier and carrier name first followed by the destination and destination name, then everything else.

```
# left join flights2 with airports
flights2 %>%
    left_join(
        # selecting only faa and name cuz that's all we need for now
        select(airports, faa, dest_name = name),
        # join where dest equals faa
        c("dest" = "faa")
        ) %>%
    # bring carrier, airline_name, dest, and dest_name up front
    select(carrier, airline_name, dest, dest_name, everything())
```

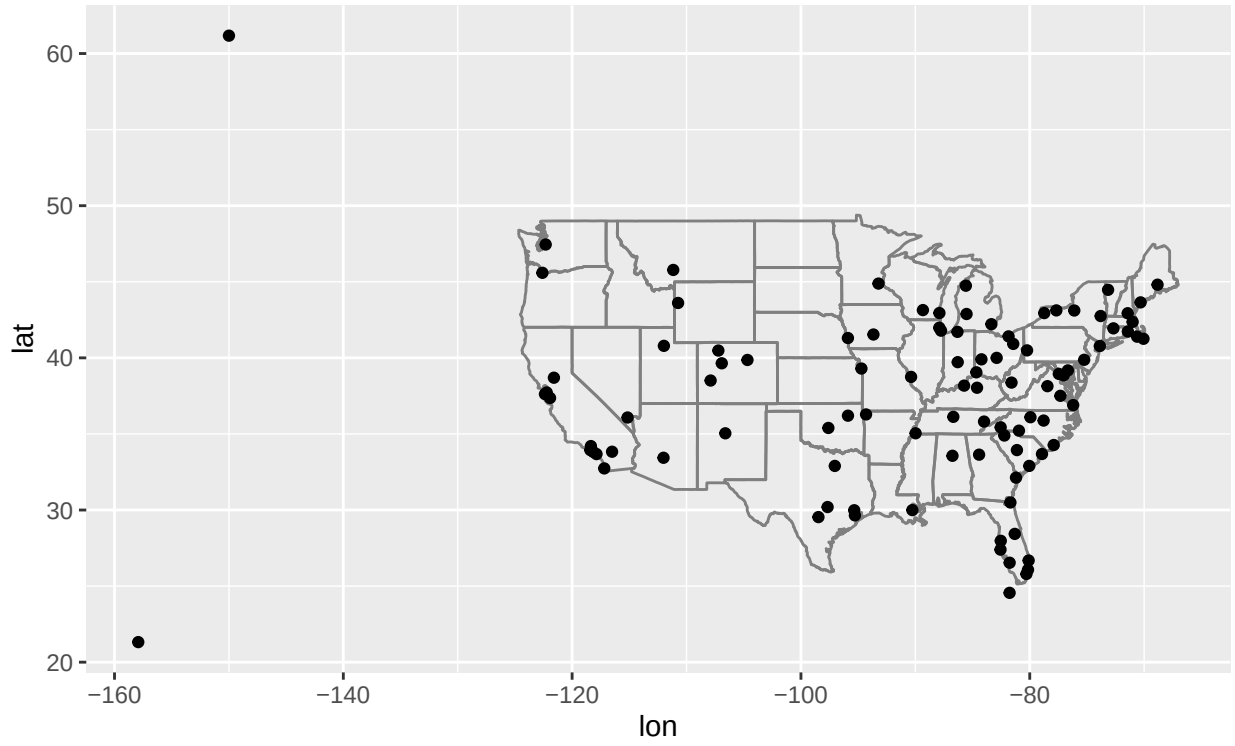
```
## # A tibble: 336,776 x 21
##    carrier airline_name             dest  dest_name    year month   day dep_time
##    <chr>   <chr>                    <chr> <chr>       <int> <int> <int>    <int>
##  1 UA      United Air Lines Inc.    IAH   George Bus~  2013     1     1      517
##  2 UA      United Air Lines Inc.    IAH   George Bus~  2013     1     1      533
##  3 AA      American Airlines Inc.   MIA   Miami Intl   2013     1     1      542
##  4 B6      JetBlue Airways          BQN   <NA>         2013     1     1      544
##  5 DL      Delta Air Lines Inc.     ATL   Hartsfield~  2013     1     1      554
##  6 UA      United Air Lines Inc.    ORD   Chicago Oh~  2013     1     1      554
##  7 B6      JetBlue Airways          FLL   Fort Laude~  2013     1     1      555
##  8 EV      ExpressJet Airlines Inc. IAD   Washington~  2013     1     1      557
##  9 B6      JetBlue Airways          MCO   Orlando In~  2013     1     1      557
## 10 AA      American Airlines Inc.   ORD   Chicago Oh~  2013     1     1      558
## # i 336,766 more rows
## # i 13 more variables: sched_dep_time <int>, dep_delay <dbl>, arr_time <int>,
## #   sched_arr_time <int>, arr_delay <dbl>, flight <int>, tailnum <chr>,
## #   origin <chr>, air_time <dbl>, distance <dbl>, hour <dbl>, minute <dbl>,
## #   time_hour <dttm>
```

Question 5. The "nycflights13" library and the code to create spatial map is provided for you. Now compute the average delay by destination, then join on the airports dataframe so you can show the spatial distribution of delays.

- Use the size or colour of the points to display the average delay for each airport.
- Add the location of the origin and destination (i.e. the lat and lon) to flights.
- Compute the average delay by destination.

Use the textbook for reference.

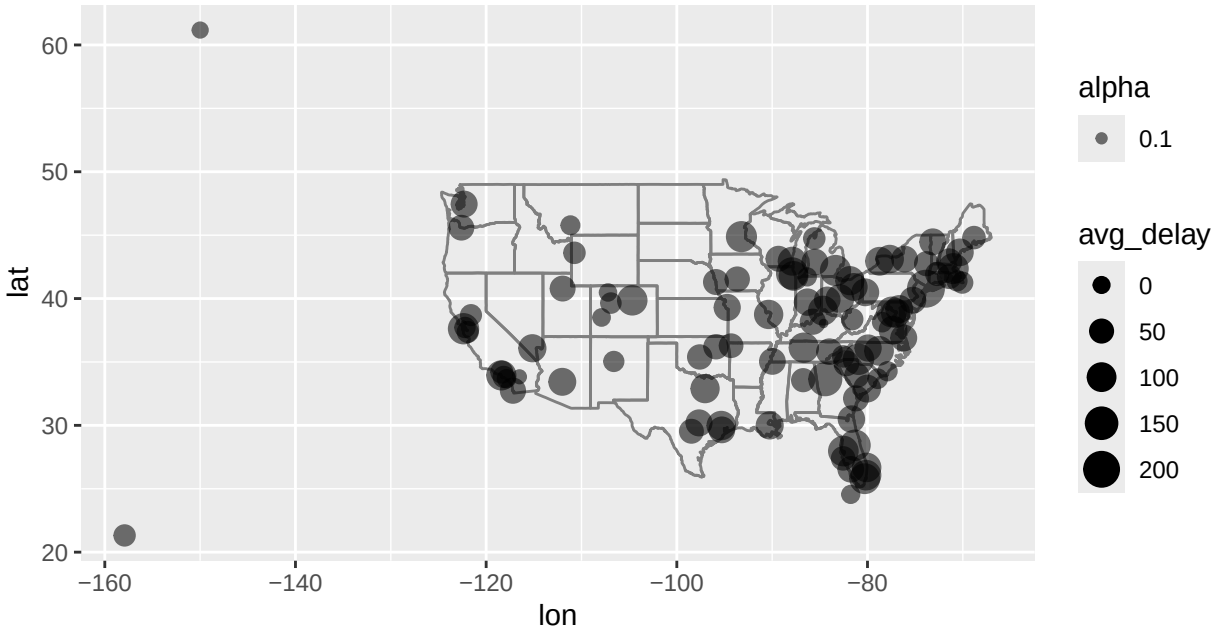
```r
# provided code
library(nycflights13)

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




```r
# my code
# average delays per airport
delays <- flights %>%
            # selecting and renaming origin code and dep delays
            select(faa = origin, delay = dep_delay) %>%
            # selecting and renaming destination code and arr delays
            # unioned together to make grouping easier
            union(select(flights, faa = dest, delay = arr_delay)) %>%
            # grouped together the origin and destination codes
            group_by(faa) %>%
            # getting the average delays and excluding NA
            summarise(avg_delay = mean(delay, na.rm = TRUE))
# here is a look at the delays numerically
delays
```

```
## # A tibble: 107 x 2
##    faa   avg_delay
##    <chr>     <dbl>
##  1 ABQ        19.2
##  2 ACK        26.0
##  3 ALB        51.3
##  4 ANC        -4.29
##  5 ATL       157.
##  6 AUS        69.2
##  7 AVL        28.0
##  8 BDL        42.3
##  9 BGR        36.5
## 10 BHM        41.9
## # i 97 more rows
```

```r
# here is a look at the delays graphically
airports %>%
    # same semi join as provided earlier to filter to same airports in both sets
    semi_join(flights, c("faa" = "dest")) %>%
    # left joining to delays to get the average delays per airport
    left_join(delays, "faa") %>%
    # plotting aesthetics, alpha added to see clustered airports better
    # size of points going off of avg delays
    ggplot(aes(lon, lat, size = avg_delay, alpha = .1)) +
    borders("state") +
    geom_point() +
    coord_quickmap()
```



Question 6. Use a set operation function to find which airport codes from flights are not in the airports dataset.

```
# find airport codes from flights not in airports
flights %>%
    anti_join(airports, c("dest" = "faa")) %>%
    select(dest) %>%
    unique()
```

```
## # A tibble: 4 x 1
##   dest
##   <chr>
## 1 BQN
## 2 SJU
## 3 STT
## 4 PSE
```

```
# I did not see any unique airport codes not found in airports from origin
```