Dplyr Assignment

Kishore Hari

2022-11-02

1 filter function

1. Find all flights that

   1. Had an arrival delay of two or more hours
   2. Flew to Houston (IAH or HOU)
   3. Were operated by United, American, or Delta
   4. Departed in summer (July, August, and September)
   5. Arrived more than two hours late, but didn't leave late
   6. Were delayed by at least an hour, but made up over 30 minutes in flight
   7. Departed between midnight and 6am (inclusive)

2. Another useful dplyr filtering helper is between(). What does it do? Can you use it to simplify the code needed to answer the previous challenges?

3. How many flights have a missing dep_time? What other variables are missing? What might these rows represent?

4. Why is NA ˆ 0 not missing? Why is NA | TRUE not missing? Why is FALSE & NA not missing? Can you figure out the general rule? (NA * 0 is a tricky counterexample!)

2 arrange function

1. How could you use arrange() to sort all missing values to the start? (Hint: use is.na()).

2. Sort flights to find the most delayed flights. Find the flights that left earliest.

3. Sort flights to find the fastest (highest speed) flights.

4. Which flights travelled the farthest? Which travelled the shortest?

3 select function

1. Brainstorm as many ways as possible to select dep_time, dep_delay, arr_time, and arr_delay from flights.

2. What happens if you include the name of a variable multiple times in a select() call?

3. What does the any_of() function do? Why might it be helpful in conjunction with this vector?

```
vars <- c("year", "month", "day", "dep_delay", "arr_delay")
```

4. Does the result of running the following code surprise you? How do the select helpers deal with case by default? How can you change that default?

```
select(flights, contains("TIME"))
```

```
## # A tibble: 336,776 x 6
##    dep_time sched_dep_time arr_time sched_arr_time air_time time_hour
##       <int>          <int>    <int>          <int>    <dbl> <dttm>
##  1      517            515      830            819      227 2013-01-01 05:00:00
##  2      533            529      850            830      227 2013-01-01 05:00:00
##  3      542            540      923            850      160 2013-01-01 05:00:00
##  4      544            545     1004           1022      183 2013-01-01 05:00:00
##  5      554            600      812            837      116 2013-01-01 06:00:00
##  6      554            558      740            728      150 2013-01-01 05:00:00
##  7      555            600      913            854      158 2013-01-01 06:00:00
##  8      557            600      709            723       53 2013-01-01 06:00:00
##  9      557            600      838            846      140 2013-01-01 06:00:00
## 10      558            600      753            745      138 2013-01-01 06:00:00
## # ... with 336,766 more rows
## # i Use ‘print(n = ...)‘ to see more rows
```

4 mutate function

1. Currently dep_time and sched_dep_time are convenient to look at, but hard to compute with because they're not really continuous numbers. Convert them to a more convenient representation of number of minutes since midnight.
2. Compare air_time with arr_time - dep_time. What do you expect to see? What do you see? What do you need to do to fix it?
3. Compare dep_time, sched_dep_time, and dep_delay. How would you expect those three numbers to be related?
4. Find the 10 most delayed flights using a ranking function. How do you want to handle ties? Carefully read the documentation for min_rank().
5. What does 1:3 + 1:10 return? Why?
6. What trigonometric functions does R provide?

5 group_by and summarise functions

1. Brainstorm at least 5 different ways to assess the typical delay characteristics of a group of flights. Consider the following scenarios:
2. A flight is 15 minutes early 50% of the time, and 15 minutes late 50% of the time.
3. A flight is always 10 minutes late.
4. A flight is 30 minutes early 50% of the time, and 30 minutes late 50% of the time.

5. 99% of the time a flight is on time. 1% of the time it's 2 hours late.
6. Which is more important: arrival delay or departure delay?
7. Come up with another approach that will give you the same output as not_cancelled %>% count(dest) and not_cancelled %>% count(tailnum, wt = distance) (without using count()).
8. Our definition of cancelled flights (is.na(dep_delay) | is.na(arr_delay)) is slightly suboptimal. Why? Which is the most important column?
9. Look at the number of cancelled flights per day. Is there a pattern? Is the proportion of cancelled flights related to the average delay?
10. Which carrier has the worst delays? Challenge: can you disentangle the effects of bad airports vs. bad carriers? Why/why not? (Hint: think about flights %>% group_by(carrier, dest) %>% summarise(n()))
11. What does the sort argument to count() do. When might you use it?

6 grouped mutates

1. Refer back to the lists of useful mutate and filtering functions. Describe how each operation changes when you combine it with grouping.
2. Which plane (tailnum) has the worst on-time record?
3. What time of day should you fly if you want to avoid delays as much as possible?
4. For each destination, compute the total minutes of delay. For each flight, compute the proportion of the total delay for its destination.
5. Delays are typically temporally correlated: even once the problem that caused the initial delay has been resolved, later flights are delayed to allow earlier flights to leave. Using lag(), explore how the delay of a flight is related to the delay of the immediately preceding flight.
6. Look at each destination. Can you find flights that are suspiciously fast? (i.e. flights that represent a potential data entry error). Compute the air time of a flight relative to the shortest flight to that destination. Which flights were most delayed in the air?
7. Find all destinations that are flown by at least two carriers. Use that information to rank the carriers.
8. For each plane, count the number of flights before the first delay of greater than 1 hour.