1.transform (exercises)

Joao Lopes

2024-10-07

Contents

1. NUMERIC VECTORS

[from https://r4ds.hadley.nz/numbers]

[see https://jrnold.github.io/r4ds-exercise-solutions/transform.html]

[see https://mine-cetinkaya-rundel.github.io/r4ds-solutions/numbers.html]

1.1. COUNTS

a) How can you use count() to count the number of rows with a missing value for a given variable?

b) Expand the following calls to count() to instead use group_by(), summarize(), and arrange():

1. flights |> count(dest, sort = TRUE)
2. flights |> count(tailnum, wt = distance)

1.2. NUMERIC TRANSFORMATION

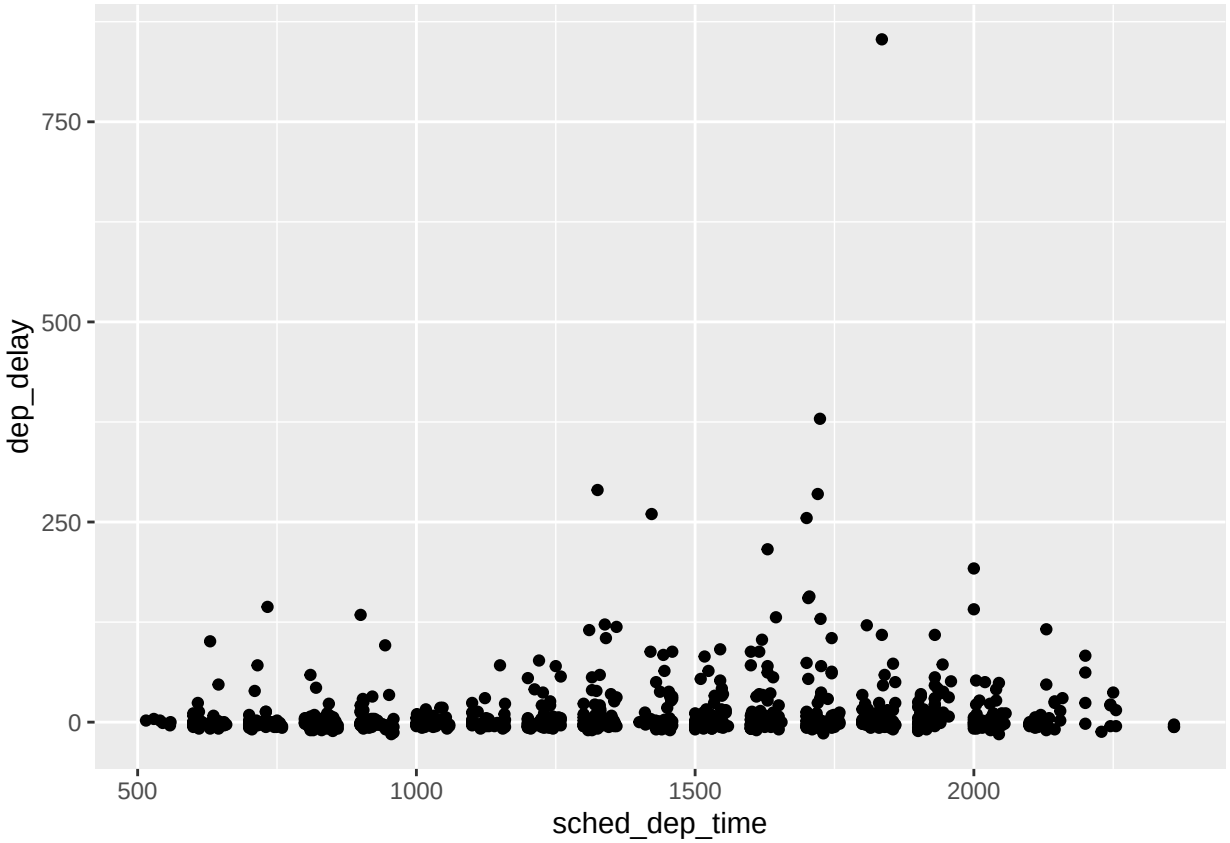
a) Explain in words what each line of the following code does:

```
flights |>
  group_by(hour = sched_dep_time %/% 100) |>
  summarize(prop_cancelled = mean(is.na(dep_time)), n = n()) |>
  filter(hour > 1) |>
  ggplot(aes(x = hour, y = prop_cancelled)) +
  geom_line(color = "grey50") +
  geom_point(aes(size = n))
```

b) Currently dep_time and sched_dep_time are convenient to look at, but hard to compute with because they're not really continuous numbers. You can see the basic problem by running the code below: there's a gap between each hour.

```
flights |>
  filter(month == 1, day == 1) |>
  ggplot(aes(x = sched_dep_time, y = dep_delay)) +
  geom_point()
```

```
## Warning: Removed 4 rows containing missing values (`geom_point()`).
```



Convert them to a more truthful representation of time (either fractional hours or minutes since midnight).

c) Round dep_time and arr_time to the nearest five minutes

1.3. GENERAL TRANSFORMATION

a) Find the 10 most delayed flights using a ranking function. How do you want to handle ties? Carefully read the documentation for min_rank().

b) Which plane (tailnum) has the worst on-time record?

c) What time of day should you fly if you want to avoid delays as much as possible?

d) What does flights |> group_by(dest) |> filter(row_number() < 4) do? What does flights |> group_by(dest) |> filter(row_number(dep_delay) < 4) do?

e) For each destination, compute the total minutes of delay. For each flight, compute the proportion of the total delay for its destination.

f) 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 average flight delay for an hour is related to the average delay for the previous hour.

```
flights |>
  mutate(hour = dep_time %/% 100) |>
  group_by(lagmonth, day, hour) |>
  summarize(
    dep_delay = mean(dep_delay, na.rm = TRUE),
    n = n(),
    .groups = "drop"
```

```
  ) |>
  filter(n > 5)
```

g) 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?

h) Find all destinations that are flown by at least two carriers. Use those destinations to come up with a relative ranking of the carriers based on their performance for the same destination.

1.4. SUMMARY STATISTICS

a) Brainstorm at least 5 different summary statistics (center, location, spread) to describe the typical delay of flights to each destination. When is mean() useful? When is median() useful? When might you want to use something else? Should you use arrival delay or departure delay?

b) Which destinations show the greatest variation in air speed?

c) Create a plot to further explore the adventures of EGE. Can you find any evidence that the airport moved locations? Can you find another variable that might explain the difference?

---

2. FACTORS

[from https://r4ds.hadley.nz/factors]

[see https://jrnold.github.io/r4ds-exercise-solutions/factors.html]

[see https://mine-cetinkaya-rundel.github.io/r4ds-solutions/factors.html]

2.1. BASICS

[no exercises]

2.2. DATASET gss_cat

a) Explore the distribution of rincome (reported income). What makes the default bar chart hard to understand? How could you improve the plot?

b) What is the most common relig in this survey? What's the most common partyid?

c) Which relig does denom (denomination) apply to? How can you find out with a table? How can you find out with a visualization?

2.3. MODIFYING FACTOR ORDER

a) For each factor in gss_cat identify whether the order of the levels is arbitrary or principled.

b) Why did moving "Not applicable" to the front of the levels move it to the bottom of the plot?

2.4. MODIFYING FACTOR LEVELS

a) How have the proportions of people identifying as Democrat, Republican, and Independent changed over time?

b) How could you collapse rincome into a small set of categories?

c) Notice there are 9 groups (excluding other) in the fct_lump example above. Why not 10? (Hint: type ?fct_lump, and find the default for the argument other_level is "Other".)

---

3. LOGICAL VECTORS

[from https://r4ds.hadley.nz/logicals]

[see https://jrnold.github.io/r4ds-exercise-solutions/vectors.html]

[see https://mine-cetinkaya-rundel.github.io/r4ds-solutions/logicals.html]

3.1. COMPARISONS

a) How does dplyr::near() work? Type near to see the source code. Is sqrt(2)^2 near 2?

b) Use mutate(), is.na(), and count() together to describe how the missing values in dep_time, sched_dep_time and dep_delay are connected.?

3.2. BOOLEAN ALGEBRA

a) Find all flights where arr_delay is missing but dep_delay is not. Find all flights where neither arr_time nor sched_arr_time are missing, but arr_delay is.

b) How many flights have a missing dep_time? What other variables are missing in these rows? What might these rows represent?

c) Assuming that a missing dep_time implies that a flight is cancelled, look at the number of cancelled flights per day. Is there a pattern? Is there a connection between the proportion of cancelled flights and the average delay of non-cancelled flights?

3.3. SUMMARIES

a) What will sum(is.na(x)) tell you? How about mean(is.na(x))?

b) What does prod() return when applied to a logical vector? What logical summary function is it equivalent to? What does min() return when applied to a logical vector? What logical summary function is it equivalent to? Read the documentation and perform a few experiments.

3.4. CONDITIONAL TRANSFORMATIONS

a) A number is even if it's divisible by two, which in R you can find out with x %% 2 == 0. Use this fact and if_else() to determine whether each number between 0 and 20 is even or odd.

b) Given a vector of days like x <- c("Monday", "Saturday", "Wednesday"), use an ifelse() statement to label them as weekends or weekdays.

c) Use ifelse() to compute the absolute value of a numeric vector called x.

d) Write a case_when() statement that uses the month and day columns from flights to label a selection of important US holidays (e.g., New Years Day, 4th of July, Thanksgiving, and Christmas). First create a logical column that is either TRUE or FALSE, and then create a character column that either gives the name of the holiday or is NA.