Homework (answer) - Week 2

Vikram Viswanath

2025-09-07

Preface

The goal of this assignment is to help you gain familiarity with data frames – think "spreadsheets" – and how to use **dplyr** functions to transform data. In this homework we are providing some code snippets to serve as "scaffolding" to help guide you through each step. As always, please come to office hours and reach out to your teaching staff if you have any questions.

NOTE: While the assignment may look long, we have already written most of the code for you in the form of "scaffolded" code that provides functions that need to be completed. In some cases you need to replace the argument `FALSE` with the correct argument. Read the questions and code comments carefully to determine what you need to fill in. Please also complete any text answers that end in ellipses (…).

We will work with the data table `flights` provided in the package **nycflights13** (details here). The data table includes all domestic flights that departed NYC in 2013.

```r
# let's take a look at the data
head(flights) # print the first six lines to fit on one page
```

```
# A tibble: 6 x 19
   year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
  <int> <int> <int>    <int>          <int>     <dbl>    <int>          <int>
1  2013     1     1      517            515         2      830            819
2  2013     1     1      533            529         4      850            830
3  2013     1     1      542            540         2      923            850
4  2013     1     1      544            545        -1     1004           1022
5  2013     1     1      554            600        -6      812            837
6  2013     1     1      554            558        -4      740            728
# i 11 more variables: arr_delay <dbl>, carrier <chr>, flight <int>,
#   tailnum <chr>, origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>,
#   hour <dbl>, minute <dbl>, time_hour <dttm>
```

1. In this data set, `arr_delay` is a variable that records the arrival delays in minutes. Negative times represent early arrivals. Use dplyr::filter to find: (1) the flights that arrived more than two hours late, and (2) the flights that arrived earlier than scheduled. What is the proportion of flights that arrived more than two hours late? What is the proportion of flights that arrived earlier than scheduled time?

```r
# Use filter to find and count flights that arrived more than two hours late
two_hour_late <- flights |>
  filter(arr_delay > 120) |>   # arrival delay more than 120 minutes
  count()

# Use filter to find and count flights that arrived earlier than scheduled
early_arr <- flights |>
  filter(arr_delay < 0) |>     # arrival delay negative = early
  count()


# Count the total number of flights
total <- count(flights)
```

A proportion of 0.03 of the flights arrived more than two hours late. A proportion of 0.56 of the flights arrived earlier than scheduled time.

2. How many flights have a missing dep_time? Look at the other variables that are also missing. What might these rows represent?

```
flights |>
  filter(is.na(dep_time)) |> # replace FALSE with your code
  count()
```

```
# A tibble: 1 x 1
      n
  <int>
1  8255
```

These rows probably represent **canceled flights**, since they have no recorded departure, arrival, or air times.

3. Use two different methods to select variables of `dep_time`, `sched_dep_time`, `dep_delay`, `arr_time`, `sched_arr_time`, `arr_delay`. Put `arr_delay` in the first column.

```
# Method 1: Use select() with column names directly
flights |>
  select(arr_delay, dep_time, sched_dep_time,
  dep_delay, arr_time, sched_arr_time)
```

```
# A tibble: 336,776 x 6
   arr_delay dep_time sched_dep_time dep_delay arr_time sched_arr_time
       <dbl>    <int>          <int>     <dbl>    <int>          <int>
 1        11      517            515         2      830            819
 2        20      533            529         4      850            830
 3        33      542            540         2      923            850
 4       -18      544            545        -1     1004           1022
 5       -25      554            600        -6      812            837
 6        12      554            558        -4      740            728
 7        19      555            600        -5      913            854
 8       -14      557            600        -3      709            723
 9        -8      557            600        -3      838            846
10         8      558            600        -2      753            745
# i 336,766 more rows
```

```
# Method 2: : Use select() with tidyselect helpers

flights |>
  select(arr_delay, any_of(c("dep_time", "sched_dep_time",
  "dep_delay", "arr_time", "sched_arr_time")))
```

```
# A tibble: 336,776 x 6
   arr_delay dep_time sched_dep_time dep_delay arr_time sched_arr_time
       <dbl>    <int>          <int>     <dbl>    <int>          <int>
 1        11      517            515         2      830            819
 2        20      533            529         4      850            830
 3        33      542            540         2      923            850
 4       -18      544            545        -1     1004           1022
 5       -25      554            600        -6      812            837
 6        12      554            558        -4      740            728
 7        19      555            600        -5      913            854
 8       -14      557            600        -3      709            723
 9        -8      557            600        -3      838            846
10         8      558            600        -2      753            745
# i 336,766 more rows
```

4a. Use dplyr::arrange to sort flights by arrival delays in descending order and print the result.

```
# sort flights by arrival delays in descending order
flights |>
    arrange(desc(arr_delay)) # replace FALSE with your code
```

```
# A tibble: 336,776 x 19
    year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
   <int> <int> <int>    <int>          <int>     <dbl>    <int>          <int>
 1  2013     1     9      641            900      1301     1242           1530
 2  2013     6    15     1432           1935      1137     1607           2120
 3  2013     1    10     1121           1635      1126     1239           1810
 4  2013     9    20     1139           1845      1014     1457           2210
 5  2013     7    22      845           1600      1005     1044           1815
 6  2013     4    10     1100           1900       960     1342           2211
 7  2013     3    17     2321            810       911      135           1020
 8  2013     7    22     2257            759       898      121           1026
 9  2013    12     5      756           1700       896     1058           2020
10  2013     5     3     1133           2055       878     1250           2215
# i 336,766 more rows
# i 11 more variables: arr_delay <dbl>, carrier <chr>, flight <int>,
#   tailnum <chr>, origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>,
#   hour <dbl>, minute <dbl>, time_hour <dttm>
```

4b. Use dplyr::slice_max to get the row with the largest arrival delay (we have already done this for you), and then use dplyr::pull to extract the value of that arrival delay. How long was the worst arrival delay?

```
# assign the longest arrival delay to worst_delay
worst_delay <- flights |>
  slice_max(arr_delay) |> # "slice" the row with the max value of arr_delay
  pull() # fill in this function with your code
```

The worst arrival delay was 2013-01-09 09:00:00 minutes.

5. Select `air_time` and `distance`. Generate a new varible `speed` that is calculated as `distance` divided by `air_time` (in miles/min). Then create a variable `mph` that contains speed in miles/hour.

```r
flights |>
  select() |>  # select variables here
  mutate(
    # create a new variable `speed`
    # create a new variable `mph`
    )
```

```
# A tibble: 336,776 x 0
```

6a. Calculate the average arrival delay by carrier.

```
flights |>
  group_by() |> # fill in this function with your code
  summarize() # fill in this function with your code
```

```
# A tibble: 1 x 0
```

6b. Which carrier has the longest average delay? Filter the row that corresponds to that carrier out of the data frame from part a.

```
flights |>
  group_by() |> # fill in this function with your code
  summarize() |> # fill in this function with your code
  slice_max(FALSE) # replace FALSE with your code
```

```
# A tibble: 1 x 0
```

7. Arriving early is better than arriving late. Based on the data, what hours of the day are on average better for flying if you want to avoid arrival delays, based on the scheduled departure hour (`hour`)?

```
# write your code here
```

The best time to fly to avoid delays is...