Homework (answer) - Week 2

Vikram Viswanath

2025-09-08

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.

```
# 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>
```

I used the knitr and kableExtra libraries to improve how my results are displayed in the report. The knitr package provides the kable() function, which lets me convert data frames or tibbles into clean outputs as tables that you see generated in this pdf.

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?

```
# Use filter to find and count flights that arrived more than two hours late
two_hour_late <- flights |>
  filter(arr_delay > 120) |>
  count(name = "n")
# Use filter to find and count flights that arrived earlier than scheduled
early_arr <- flights |>
  filter(arr_delay < 0) |>
  count(name = "n")
# Count the total number of flights
total <- flights |> count(name = "n")
```

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)) |>
  count(name = "n") |>
  kable(digits = 0, col.names = c("Rows with Missing dep_time"))
```

Rows with Missing dep_time
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) |>
  head(10) |>
  kable(digits = 0, col.names =
  c("arr_delay","dep_time","sched_dep_time","dep_delay","arr_time","sched_arr_time"))
```

arr_delay	dep_time	sched_dep_time	dep_delay	arr_time	sched_arr_time
11	517	515	2	830	819
20	533	529	4	850	830
33	542	540	2	923	850
-18	544	545	-1	1004	1022
-25	554	600	-6	812	837
12	554	558	-4	740	728
19	555	600	-5	913	854
-14	557	600	-3	709	723
-8	557	600	-3	838	846
8	558	600	-2	753	745

```
# 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"))) |>
  head(10) |>
  kable(digits = 0, col.names = c("arr_delay","dep_time","sched_dep_time",
  "dep_delay","arr_time","sched_arr_time"))
```

arr_delay	dep_time	sched_dep_time	dep_delay	arr_time	sched_arr_time
11	517	515	2	830	819
20	533	529	4	850	830
33	542	540	2	923	850
-18	544	545	-1	1004	1022
-25	554	600	-6	812	837

arr_delay	dep_time	sched_dep_time	dep_delay	arr_time	sched_arr_time
12	554	558	-4	740	728
19	555	600	-5	913	854
-14	557	600	-3	709	723
-8	557	600	-3	838	846
8	558	600	-2	753	745

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)) |>
  head(6) |>
  select(year, month, day, dep_time, arr_time,
  arr_delay, carrier, flight, dest) |>
  kable(digits = 0) |>
  kable_styling(font_size = 8)
```

year	month	day	dep_time	arr_time	arr_delay	carrier	flight	dest
2013	1	9	641	1242	1272	HA	51	HNL
2013	6	15	1432	1607	1127	MQ	3535	CMH
2013	1	10	1121	1239	1109	MQ	3695	ORD
2013	9	20	1139	1457	1007	AA	177	SFO
2013	7	22	845	1044	989	MQ	3075	CVG
2013	4	10	1100	1342	931	DL	2391	TPA

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(arr_delay) # fill in this function with your code
```

The worst arrival delay was 1272 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.

```
flights |>
  select(air_time, distance) |>
  mutate(
    speed = distance / air_time,
    mph   = speed * 60
  ) |>
  slice_head(n = 20) |>
  kable(digits = 2, col.names = c("Air Time (min)",
  "Distance (miles)", "Speed (mi/min)", "Speed (mph)"))
```

Air Time (min)	Distance (miles)	Speed (mi/min)	Speed (mph)
227	1400	6.17	370.04
227	1416	6.24	374.27
160	1089	6.81	408.38
183	1576	8.61	516.72
116	762	6.57	394.14
150	719	4.79	287.60
158	1065	6.74	404.43
53	229	4.32	259.25
140	944	6.74	404.57
138	733	5.31	318.70
149	1028	6.90	413.96
158	1005	6.36	381.65
345	2475	7.17	430.43
361	2565	7.11	426.32
257	1389	5.40	324.28
44	187	4.25	255.00
337	2227	6.61	396.50
152	1076	7.08	424.74
134	762	5.69	341.19
147	1023	6.96	417.55

6a. Calculate the average arrival delay by carrier.

```
flights |>
  group_by(carrier) |>
  summarize(avg_arr_delay = mean(arr_delay, na.rm = TRUE)) |>
  kable(
    digits = 2,
    col.names = c("Carrier", "Avg. Arrival Delay (min)")
  )
```

Carrier	Avg. Arrival Delay (min)
9E	7.38
AA	0.36
AS	-9.93
B6	9.46
DL	1.64
EV	15.80
F9	21.92
FL	20.12
HA	-6.92
MQ	10.77
OO	11.93
UA	3.56
US	2.13
VX	1.76
WN	9.65
YV	15.56

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(carrier) |> # fill in this function with your code
  summarize(avg_arr_delay = mean(arr_delay, na.rm = TRUE)) |>
  slice_max(avg_arr_delay) |>  # replace FALSE with your code
  kable(
    digits = 2,
    col.names = c("Carrier", "Avg. Arrival Delay (min)")
  )
```

Carrier	Avg. Arrival Delay (min)
F9	21.92

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

best_hours <- flights |>
  group_by(hour) |>
  summarize(
    avg_arr_delay = mean(arr_delay, na.rm = TRUE),
    n = n()
  ) |>
  arrange(avg_arr_delay)

best_hours |>
  kable(
    digits = 2,
    col.names = c("Hour", "Avg. Arrival Delay (min)", "Number of Flights")
  )
```

Hour	Avg. Arrival Delay (min)	Number of Flights
7	-5.30	22821
5	-4.80	1953
6	-3.38	25951
9	-1.45	20312
8	-1.11	27242
10	0.95	16708
11	1.48	16033
12	3.49	18181
13	6.54	19956
14	9.20	21706
23	11.76	1061
15	12.32	23888
16	12.60	23002
18	14.79	21783
22	15.97	2639
17	16.04	24426
19	16.66	21441
20	16.68	16739
21	18.39	10933

Hour	Avg. Arrival Delay (min)	Number of Flights
1	NaN	1

The best time to fly to avoid delays is typically during the early morning hours. Based on the data, the hours with the lowest average arrival delays are 7, 5, 6.