Reading Set 2

Kauila Tengan

Due by 10pm ET on Monday

Reading Set Information

A more thorough reading and light practice of the textbook reading prior to class allows us to jump into things more quickly in class and dive deeper into topics. As you actively read the textbook, you will work through the Reading Sets to help you engage with the new concepts and skills, often by replicating on your own the examples covered in the book.

These should be completed on your own without help from your peers. While most of our work in this class will be collaborative, it is important each individual completes the active readings. The problems should be straightforward based on the textbook readings, but if you have any questions, feel free to ask me!

GitHub Workflow

1. Before editing this file, verify you are working on the copy saved in *your* repo for the course (check the filepath and the project name in the top right corner).
2. Before editing this file, make an initial commit of the file to your repo to add your copy of the problem set.
3. Change your name at the top of the file and get started!
4. You should *save, knit, and commit* the .Rmd file each time you've finished a question, if not more often.
5. You should occasionally *push* the updated version of the .Rmd file back onto GitHub. When you are ready to push, you can click on the Git pane and then click **Push**. You can also do this after each commit in RStudio by clicking **Push** in the top right of the *Commit* pop-up window.
6. When you think you are done with the assignment, save the pdf as "*Name_thisfilename_date*.pdf" (it's okay to leave out the date if you don't need it) before committing and pushing (this is generally good practice but also helps me in those times where I need to download all student homework files).

Gradescope Upload

For each question (e.g., 3.1), allocate all pages associated with the specific question. If your work for a question runs onto a page that you did not select, you may not get credit for the work. If you do not allocate *any* pages when you upload your pdf, you may get a zero for the assignment.

You can resubmit your work as many times as you want before the deadline, so you should not wait until the last minute to submit some version of your work. Unexpected delays/crises that occur on the day the assignment is due do not warrant extensions (please submit whatever you have done to receive partial credit).

Problem 1 **NYC Flights** In Section 5.1, the `flights` and `carrier` tables within the `nycflights13` package are joined together.

1.1 Recreate the `flights_joined` dataset from Section 5.1, being sure to *glimpse* the data in the Console to verify the join worked.

```
#glimpse(flights)
flights_joined <- flights %>%
  inner_join(airlines, by = c("carrier" = "carrier"))
glimpse(flights_joined)
```

```
Rows: 336,776
Columns: 20
$ year           <int> 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2~
$ month          <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1~
$ day            <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1~
$ dep_time       <int> 517, 533, 542, 544, 554, 554, 555, 557, 557, 558, 558, ~
$ sched_dep_time <int> 515, 529, 540, 545, 600, 558, 600, 600, 600, 600, 600, ~
$ dep_delay      <dbl> 2, 4, 2, -1, -6, -4, -5, -3, -3, -2, -2, -2, -2, -2, -1~
$ arr_time       <int> 830, 850, 923, 1004, 812, 740, 913, 709, 838, 753, 849,~
$ sched_arr_time <int> 819, 830, 850, 1022, 837, 728, 854, 723, 846, 745, 851,~
$ arr_delay      <dbl> 11, 20, 33, -18, -25, 12, 19, -14, -8, 8, -2, -3, 7, -1~
$ carrier        <chr> "UA", "UA", "AA", "B6", "DL", "UA", "B6", "EV", "B6", "~
$ flight         <int> 1545, 1714, 1141, 725, 461, 1696, 507, 5708, 79, 301, 4~
$ tailnum        <chr> "N14228", "N24211", "N619AA", "N804JB", "N668DN", "N394~
$ origin         <chr> "EWR", "LGA", "JFK", "JFK", "LGA", "EWR", "EWR", "LGA",~
$ dest           <chr> "IAH", "IAH", "MIA", "BQN", "ATL", "ORD", "FLL", "IAD",~
$ air_time       <dbl> 227, 227, 160, 183, 116, 150, 158, 53, 140, 138, 149, 1~
$ distance       <dbl> 1400, 1416, 1089, 1576, 762, 719, 1065, 229, 944, 733, ~
$ hour           <dbl> 5, 5, 5, 5, 6, 5, 6, 6, 6, 6, 6, 6, 6, 6, 6, 5, 6, 6, 6~
$ minute         <dbl> 15, 29, 40, 45, 0, 58, 0, 0, 0, 0, 0, 0, 0, 0, 0, 59, 0~
$ time_hour      <dttm> 2013-01-01 05:00:00, 2013-01-01 05:00:00, 2013-01-01 0~
$ name           <chr> "United Air Lines Inc.", "United Air Lines Inc.", "Amer~
```

1.2 Now, starting from `flights_joined`, create a new dataset `flights_short` that **(1)** creates a new variable, `distance_km`, which is distance in kilometers (note that 1 mile is about 1.6 kilometers); **(2)** keeps only the variables: `name`, `flight`, `arr_delay`, and `distance_km`; and **(3)** keeps only observations where the distance is less than 500 kilometers.

```
flights_short <- flights_joined %>%
  mutate(distance_km = distance*(1.6)) %>%
  select(name, flight, arr_delay, distance_km)

flights_short <- filter(flights_short, distance_km <= 500)

glimpse(flights_short)
```

```
Rows: 54,921
Columns: 4
$ name        <chr> "ExpressJet Airlines Inc.", "JetBlue Airways", "Southwest ~
$ flight      <int> 5708, 1806, 4646, 4144, 1002, 102, 20, 44, 1172, 1838, 27,~
$ arr_delay   <dbl> -14, -4, -19, 12, -10, 5, -1, 4, -19, -22, -14, -13, 851, ~
$ distance_km <dbl> 366.4, 299.2, 296.0, 339.2, 299.2, 481.6, 422.4, 334.4, 32~
```

1.3 Using the functions introduced in Section 4.1.4, compute the number of flights (call this `N`), the average arrival delay (call this `avg_arr_delay`), and the average distance in kilometers (call this `avg_dist_km`) among these flights with distances less than 500 km (i.e. working off of `flights_short`), grouping by the carrier name. Sort the results in descending order based on `avg_arr_delay`. Save the results in a tibble object called `delay_summary`, and display the table.

```
delay_summary <- tibble(flights_short %>%
  drop_na() %>%
  group_by(name) %>%
  summarize(
    N = n(),
    avg_arr_delay = mean(arr_delay),
    avg_dist_km = mean(distance_km)
  ) %>%
  arrange(desc(avg_arr_delay)))

delay_summary
```

```
# A tibble: 11 x 4
   name                        N avg_arr_delay avg_dist_km
   <chr>                   <int>         <dbl>       <dbl>
 1 Mesa Airlines Inc.        286        18.0          360.
 2 ExpressJet Airlines Inc. 14753        15.6          373.
 3 Envoy Air                 2741        11.0          351.
 4 JetBlue Airways          13443         8.66         385.
 5 Endeavor Air Inc.         6144         6.82         339.
 6 Southwest Airlines Co.     200         4.92         272.
 7 United Air Lines Inc.     3307         4.09         320.
 8 SkyWest Airlines Inc.        1         3            366.
 9 US Airways Inc.           9093         2.22         308.
10 American Airlines Inc.    1428         1.88         299.
11 Delta Air Lines Inc.      1201        -0.643        325.
```

```
#Figure out the tibble object and call it delay_summary & display table
```

1.4 Rename the four columns in the `delay_summary` data table to `Airline`, `"Total flights under 500 km"`, `"Average arrival delay (mins)"` and `"Average distance (km)"`, respectively, then use `kable(booktabs = TRUE, digits = 0)` to make the final table output in the pdf close to publication quality.

```
delay_summary <- delay_summary %>%
  rename("Airline" = name,
       "Total flights under 500 km" = N,
        "Average arrival delays (mins)" = avg_arr_delay,
        "Average distance (km)" = avg_dist_km
        ) %>%
kable(booktabs = TRUE, digits = 1)

delay_summary
```

Airline	Total flights under 500 km	Average arrival delays (mins)	Average distance (km)
Mesa Airlines Inc.	286	18.0	360.4
ExpressJet Airlines Inc.	14753	15.6	373.1
Envoy Air	2741	11.0	350.7
JetBlue Airways	13443	8.7	384.8
Endeavor Air Inc.	6144	6.8	338.9
Southwest Airlines Co.	200	4.9	272.3
United Air Lines Inc.	3307	4.1	319.7
SkyWest Airlines Inc.	1	3.0	366.4
US Airways Inc.	9093	2.2	308.3
American Airlines Inc.	1428	1.9	299.2
Delta Air Lines Inc.	1201	-0.6	324.9

Problem 2 **Baby names**

2.1 Working with the `babynames` data in the **babynames** package, create a dataset `recent_names` that only includes years 2000 to 2017.

```
recent_names <- babynames %>%
  filter(year %in% 2000:2017)
```

2.2 Following the code presented in Section 6.2.5, create a dataset called `recentnames_summary` that summarizes the total number of people in recent history (years 2000 to 2017) with each name, grouped by sex.

```
recentnames_summary <- recent_names %>%
  group_by(name, sex) %>%
  summarize(total = sum(n))
```

2.3 Now, following the fourth and fifth code chunks presented in Section 6.2.5, reshape or *pivot* the summary data from *long* format to *wide* format. Only keep observations where more than 10,000 babies have been named in each sex (`M` and `F`), and find the smaller of the two ratios `M / F` and `F / M` to identify the top three sex-balanced names (and only the top three!). Save the wide data as `recentnames_balanced_wide`. Display the table.

```
recentnames_balanced_wide <- recentnames_summary %>%
  pivot_wider(
    names_from = sex,
    values_from = total,
    values_fill = 0
  )

#head(recentnames_balanced_wide, 3)

recentnames_balanced_wide %>%
  filter(M > 10000, F > 10000) %>%
  mutate(ratio = pmin(M / F, F/ M)) %>%
  arrange(desc(ratio)) %>%
  head(3)
```

```
# A tibble: 3 x 4
# Groups:   name [3]
  name        M     F ratio
  <chr>   <int> <int> <dbl>
1 Justice 11267 10947 0.972
2 Skyler  22154 17120 0.773
3 Quinn   19080 25022 0.763
```

2.4 Finally, use `pivot_longer()` to put the dataset back into *long* form. Call this dataset `recentnames_balanced` and display the table. Why are the number of observations in `recentnames_balanced_wide` different from that in `recentnames_summary` from Problem 2.2?

> I noticed that we had filtered out and only taken the names with greater than 10,000 individuals for males and females. That is why there are ~67,000 observations for recentnames_balanced_wide versus >73,000 observations for recentnames_summary.

```
recentnames_balanced <- recentnames_balanced_wide %>%
  pivot_longer(-name, names_to = "sex", values_to = "total")
```

Problem 3 **Ethical conundrums** Each subsection of Section 8.4 discusses an ethical scenario and ends with one or more questions. Choose one of the scenarios provided to reflect on, and *in one paragraph or less* respond to the question(s) posed with your initial thoughts. Please identify the scenario for reference (e.g. "8.4.1 The chief executive officer").

Responding to 8.4.1. This reminds me of a story Shu-Min told us as she worked for her college as a data scientist. I had a similar experience working for an environmental engineering consulting firm this summer and being tasked with analyzing 10 years worth of water sampling and create graphs for the Massachusetts Department of Public Health. The consultant should obviously respond, "no this is unethical and has lots of potential to harm many people in the future if we choose to disseminate misinformation to the public just for profit." The reality however is that data consultant may get fired and lose their entirely livelihood for standing up for what is right. The way modern western science works today is that research is funded by grants. Millions of dollars flooding to fund long-term experiments or studies. The reality is that you produce results or lose your job. It is a tough situation and that is why people must be ethical.