# Practice Prelim 1

## AEM 2850 / AEM 5850

### Preface

The goal of this prelim is to assess your facility with key data wrangling tasks we covered in the first five weeks of the course.

We will work with multiple data sets, most of which are already loaded and all of which are available in the working directory of the project (see the Files pane on the lower right).

### Instructions

- You must complete Prelim 1 **in person** in Warren 150 during class
- Prelim 1 is open internet, but **do not communicate with classmates**
- Do not use packages outside the `tidyverse` packages we have already loaded for you (penalties may apply)
- When done, **upload BOTH your .qmd and .pdf files** to canvas

### Additional notes

- The prelim is 100 points total, and each question states the number of points it is worth (*this is not true on the practice prelim but will be true on the actual prelim*)
- **Render early and often** to avoid wasting time sorting out what code needs debugging
- We will give partial credit if your answers are incomplete, especially if you provide comments or text that describes the logic of what you *would* do if you had more time
- If you have trouble rendering your document, do not delete your work in progress code. That will make it hard for us to give you partial credit. Instead, you can:
  - Comment out problematic code using `#` or keyboard shortcut Cntrl/Cmd-Shift-C
  - Replace `{r}` with `{r, eval = FALSE}` at the top of the relevant code chunk
  - Ask questions!

**1. The data frame `tidy_us_registrations` contains information on vehicle registrations in the U.S. Use it top determine what make - model combination has the most registrations in the data.**

```
tidy_us_registrations
```

```
# A tibble: 70,278 x 5
   make       model     model_year count   age
   <chr>      <chr>          <dbl> <dbl> <dbl>
 1 ALFA ROMEO *UNKNOWN*       1973 34893    33
 2 ALFA ROMEO *UNKNOWN*       1974   985    32
 3 ALFA ROMEO *UNKNOWN*       1975     0    31
 4 ALFA ROMEO *UNKNOWN*       1976   149    30
 5 ALFA ROMEO *UNKNOWN*       1977     0    29
 6 ALFA ROMEO *UNKNOWN*       1978    55    28
 7 ALFA ROMEO *UNKNOWN*       1979 17053    27
 8 ALFA ROMEO *UNKNOWN*       1980     0    26
 9 ALFA ROMEO *UNKNOWN*       1981     2    25
10 ALFA ROMEO *UNKNOWN*       1982     0    24
# i 70,268 more rows
```

**The make and model with the most registrations is...**

**2. Again using `tidy_us_registrations`, if we consider all different versions of the Ford F-150 to be a single model, rather than different models, would the answer to the previous question change? Why or why not?**

...

**3. Use `case_when()` to make a table of the share of registered vehicles that fall into the following age groups: `0-2 years old`, `3-5 years old`, `6-10 years old`, `11-15 years old`, and `16+ years old`. Use those age ranges in the table so the results are clear to the reader, and put them in order from youngest to oldest. Present the numbers as percentage points, so that they sum to 100.**

## Product reviews

### 4a. The data frame `tidy` contains product reviews from Amazon. How many different products are there, based on different values of `name`?

```
tidy
```

```
# A tibble: 1,585 x 9
   brand  name         price reviews.date reviews.doRecommend reviews.numHelpful
   <chr>  <chr>        <dbl> <date>       <lgl>                             <dbl>
 1 Amazon Kindle Pape~  140. 2015-08-08   NA                                  139
 2 Amazon Kindle Pape~  140. 2015-09-01   NA                                  126
 3 Amazon Kindle Pape~  140. 2015-07-20   NA                                   69
 4 Amazon Kindle Pape~  140. 2017-06-16   NA                                    2
 5 Amazon Kindle Pape~  140. 2016-08-11   NA                                   17
 6 Amazon Kindle Pape~  140. 2015-07-08   NA                                   NA
 7 Amazon Kindle Pape~  140. 2015-09-01   NA                                   NA
 8 Amazon Kindle Pape~  140. 2015-07-03   NA                                   NA
 9 Amazon Kindle Pape~  140. 2015-08-08   NA                                   NA
10 Amazon Kindle Pape~  140. 2015-07-20   NA                                   NA
# i 1,575 more rows
# i 3 more variables: reviews.rating <dbl>, reviews.text <chr>,
#   reviews.title <chr>
```

…

### 4b. What is the most expensive product in `tidy`?

…

**5a. Let's use `tidy` to look at reviews and ratings. Which product has the largest number of reviews that others marked as helpful?**

...

**5b. Which product has the lowest average rating?**

...

**6. *Logic, strings, and regular expressions:* Let's zoom into products with "Kindle" in the name, but excluding cover and charger accessories. Use `tidy` to generate a logical variable/column in the data that indicates whether each product is a `tablet`, based on whether "Fire" is in a product's name. For each of the two categories – tablet or not – compute the number of products and average rating. Present the information in a table with two rows and three columns. Did the tablets receive higher or lower ratings than non-tablets (i.e., e-readers)?**

...

## Taxi cabs

**7a. *Joins:* Now we'll use data on taxi trips in `trips` and taxi locations in `locations`. Join them based on the pick up location (`PULocation`) from `trips` and the location ID (`LocationID`) from `locations`, taking care to retain all rows from the trip data and no rows from the location data that do not match with the trips data. How many rows are in the joined data?**

```
trips
```

```
# A tibble: 56,443 x 16
   pickup_datetime     dropoff_datetime    passenger_count trip_distance
   <dttm>              <dttm>                        <dbl>         <dbl>
 1 2021-01-28 00:05:10 2021-01-28 00:30:05               1         15.2
 2 2021-01-28 00:01:31 2021-01-28 00:07:04               2          1.4
 3 2021-01-28 00:22:05 2021-01-28 00:30:23               1          1.12
 4 2021-01-28 00:00:16 2021-01-28 00:14:51               1          3.3
 5 2021-01-28 00:24:26 2021-01-28 00:27:27               2          0.7
 6 2021-01-28 00:14:16 2021-01-28 00:18:35               1          0.7
 7 2021-01-28 00:23:51 2021-01-28 00:39:43               1          9.9
 8 2021-01-28 00:57:33 2021-01-28 01:13:33               5          3.61
 9 2021-01-28 00:49:08 2021-01-28 00:53:24               2          0.83
10 2021-01-28 00:17:51 2021-01-28 00:42:35               1          6.15
# i 56,433 more rows
# i 12 more variables: RatecodeID <dbl>, PULocationID <dbl>,
#   DOLocationID <dbl>, payment_type <dbl>, fare_amount <dbl>, extra <dbl>,
#   mta_tax <dbl>, tip_amount <dbl>, tolls_amount <dbl>,
#   improvement_surcharge <dbl>, total_amount <dbl>, congestion_surcharge <dbl>
```

```
locations
```

```
# A tibble: 265 x 4
   LocationID Borough       Zone                     service_zone
        <dbl> <chr>         <chr>                    <chr>
 1          1 EWR           Newark Airport           EWR
 2          2 Queens        Jamaica Bay              Boro Zone
 3          3 Bronx         Allerton/Pelham Gardens  Boro Zone
 4          4 Manhattan     Alphabet City            Yellow Zone
 5          5 Staten Island Arden Heights            Boro Zone
 6          6 Staten Island Arrochar/Fort Wadsworth  Boro Zone
```

```
 7          7 Queens        Astoria                Boro Zone
 8          8 Queens        Astoria Park           Boro Zone
 9          9 Queens        Auburndale             Boro Zone
10         10 Queens        Baisley Park           Boro Zone
# i 255 more rows
```

…

**7b. Are there any rows in `trips` with pick up locations that do not match a location in `locations`? If so, how many?**

…

**7c. Are there any rows in the `locations` data that do not match the pick up location of at least one taxi trip? If so, how many?**

…

**8. Based on the joined data from 7a, how many trips originated in each borough? Please present the results as a small data frame with the same number of rows as there are boroughs in NYC.**

**9. Now join the data to find the borough where the passenger was picked up as well as the borough where the passenger was dropped off (`DOLocationID`). Make a table of the most common three borough-to-borough trip combinations that did *NOT* start and end in the same borough. Please include the number of trips for each combination in the table.**