Lecture 3: Factors, Dates, and Special Values

Problem Set for Lecture 3

These questions will assist in bolstering your understanding of the material in Lecture 3. Emphasis should be placed on having the correct code/output as well as communication. The deliverable should be a knitted RMarkdown document to pdf without any code running off the page. You will be able to present these in office hours as an oral assessment interview along with the problem set for Lecture 2 and Lecture 4.

***** First, make sure you have read the Lecture Material's writeup *****

1. Answer all of the following questions relating to Date/Times **without** using any functions in the `lubridate` library:

   (a) Mount St. Mary's University was founded on "June 16, 1808" and the moon landing was on the "20th of July, 1969". How many days separate the two? [*use the exact same formats I used above when inputting the date*]

   (b) The first airplane flight by the Wright Brothers was on "December 17th, 1903 at 10:35am" while the moon landing was in "1969- July 20 at 9:32am". How many days separated the two monumental flights? What about seconds? What about weeks? [*use the exact same formats I used above when inputting the date*]

   (c) You just found that it was ____ time between the two flights. If another monumental travel breakthrough should have/will occur after the same length of time after the Moon landing, when should/will that be (make sure to include seconds)? How many days away is that new date from our class on Thursday 2/5 at 11:47am?

2. Answer the following questions relating to Date/Times by using the `flights` dataset within the `nycflights13` library (you may need to install the package).

   (a) Create a new date/time variable called `scheduled_time` that represents the scheduled departure time for each flight. Use the `paste()` function to accomplish this task to incorporate `year`, `month`, `day`, `hour`, and `minute` into a single variable.

   (b) Convert `scheduled_time` to `POSIXlt` and create a new variable based on the components of the output to capture the day of the week (Sunday = 0, Monday = 1, etc.). Do not capture the day of the week using the `wday()` function, rather manually grab a `wday` component from the POSIX list time components.

   (c) Convert the new variable you created into an ordered factor with the labels changed to the name of the day instead of the number and with the labels showing Sunday as the smallest day and Saturday as the largest day.

   (d) Using the `table()` function, determine which day of the week had the least number of scheduled flights.

3. Answer the following questions relating to special values by using the `weather` dataset within the `nycflights13` library.

   (a) Create a new vector called `pressure_clean` that contains the `pressure` column from `weather`. Determine how many missing values occur in the vector and what proportion of the observations are missing.

   (b) Determine what the mean of this new vector (you should return a quantitative value).

   (c) First, round the `wind_speed` variable to the nearest whole number. Then, create a different vector called `wind_speed_metric`, which calculates the metric as:

$$\sqrt{\frac{\text{wind speed}}{\text{median(wind speed) - wind speed}}}$$

   (d) Using the `wind_speed_metric` variable, determine how many of the values are purely missing (not counting the NaN values), how many of the values are "not a number", and how many of the values are infinite

   (e) Calculate the standard deviation of the `wind_speed_metric` after converting all "not a numbers" to be missing and after converting all the infinite values to be 2.