STA035B Homework 2, due: 2/1, 9pm

Spencer Frei

Instructions

Upload a PDF file, named with your UC Davis email ID and homework number (e.g., sfrei_hw2.pdf), to Gradescope (accessible through Canvas). You will give the commands to answer each question in its own code block, which will also produce output that will be automatically embedded in the output file. All code used to answer the question must be supplied, as well as written statements where appropriate.

All code used to produce your results must be shown in your PDF file (e.g., do not use `echo = FALSE` or `include = FALSE` as options anywhere). Rmd files do not need to be submitted, but may be requested by the TA and must be available when the assignment is submitted.

Students may choose to collaborate with each other on the homework, but must clearly indicate with whom they collaborated.

Problem 1

```
library(nycflights13)
```

Consider the `weather` dataset (comes when you load `nycflights13` library), which has columns: "origin", "year", "month", "day", "hour", "temp", "dewp", "humid", "wind_dir", "wind_speed", "wind_gust", "precip", "pressure", "visib", and "time_hour". We show the first few rows and columns below.

```
weather
```

```
# A tibble: 26,115 x 15
   origin  year month   day  hour  temp  dewp humid wind_dir wind_speed
   <chr>  <int> <int> <int> <int> <dbl> <dbl> <dbl>    <dbl>      <dbl>
 1 EWR     2013     1     1     1  39.0  26.1  59.4      270       10.4
 2 EWR     2013     1     1     2  39.0  27.0  61.6      250        8.06
 3 EWR     2013     1     1     3  39.0  28.0  64.4      240       11.5
 4 EWR     2013     1     1     4  39.9  28.0  62.2      250       12.7
 5 EWR     2013     1     1     5  39.0  28.0  64.4      260       12.7
...
```

(a) Provide code which computes the average precipitation per origin per month, removing any missing values. Then filter the resulting tibble so that only those origin-month pairs with the 5 highest average precipitation remain.

```
# code here
```

(b) Do a similar calculation: compute the average precipitation per origin per **week number** (i.e., Jan 1 - Jan 7 is week 1, Jan 8 - 15 is week 2, etc.) removing any missing values. Then filter the resulting tibble so that only those origin-month pairs with the 5 highest average precipitation remain. Note that `weather` does not have a week number so you need to create this yourself—think of what mathematical operations allow for you to find the week number, and look back at the slides on dates/times.

```
# code here
```

Problem 2

- Suppose we have a tibble of the following form:

```
df <- tribble(
  ~name, ~date_of_birth, ~favorite_food,
  "Will", "1995-09-01", "Tacos",
  "Angela", "1993-01-02", "Sushi",
  "Ana", "1994-11-20", "Italian"
)
```

(a) Provide R code which adds the following columns:

- `year`, a number, indicating the year of birth
- `month`, a string, the month (fully spelled out, i.e. January) of birth
- `day`, a number, indicating the day of the month the person was born

The resulting tibble should have 6 columns, the 3 original ones plus these 3 new ones.

```
# code here
```

(b) Provide R code which adds the following column:

- `ten_years_later`: a date time, indicating a the date corresponding to ten years after the person's birth

The resulting tibble should have 4 columns, the original 3 plus this new one.

```
# code here
```

(c) Provide R code which adds the following column:

- `age_when_obama_born`: an **integer**, indicating how many years old the person was when Barack Obama was born (August 4, 1961)
  - Be sure it is an integer and there are no decimal places!
  - The resulting tibble should have 4 columns, the original 3 plus this new one. No extra columns!

```
# code here
```

Problem 3

- Suppose we have the following tibble:

```
entered_data <- tribble(
  ~id, ~entry,
  0, "Arthur_1985-09-01_Present",
  1, "Zack_1983-01-02_Absent",
  2, "Pat_1984-11-20_Present",
)
```

Provide R code which parses the `entry` column and creates three new columns:

- `name`, a string, indicating the person's name (preceding the first underscore),
- `date_of_birth`, a date time indicating the person's date of birth,
- `day_can_vote`, a date time indicating the day when the person is 18 years old.

The resulting tibble should have 5 columns: `id` and `entry` in addition to these 3 new columns.

```
# code here
```