Homework 2

Instructions

Complete the following problems and submit to Gradescope your work as a file named `hw2.Rmd`.

A file with the wrong name or a bug will prevent the autograder from being able to run your submission. Check that your filename is correct, and check for bugs clearing your memory and then running your file from start to finish. It's also a good idea to check that your file "knits."

Do not use any loops!

Problem 1

A commuter budgets 35 minutes each morning for her commute to work regardless of the time she leaves home. One week she left home at exactly 8am every day and recorded the amount of time it took for her to get to her office. The next week she left home at exactly 8:30am every day and recorded the amount of time it took for her to get to her office.

a. Create and store a `matrix` called `commutes` with a column representing each week containing the following information:
   - On Monday of the two weeks, she arrived in 29 and 27 minutes, respectively
   - On Tuesday of the two weeks, she arrived in 38 and 32 minutes, respectively
   - On Wednesday of the two weeks, she arrived in 31 and 44 minutes, respectively
   - On Thursday of the two weeks, she arrived in 34 and 37 minutes, respectively
   - On Friday of the two weeks, she arrived in 36 and 38 minutes, respectively

b. Label the rows of `commutes` with the corresponding day of the week (title case) and label the columns with the labels `Week1` and `Week2`.

c. Compare her commute on each day of the week and determine whether she arrived faster leaving at 8:30am or not each day. Create a `logical vector` called `secondIsFaster` where a `TRUE` means that the second week time was faster.

d. Determine the average commute time for each weekday over the two week period. Call the result `aveTimes`.

e. Determine how many minutes over her budgeted time each commute lasted (early arrivals should be negative values) and save these values in a `matrix` called `myDiff`.

f. Determine the average difference over each week and store this information in the variable `aveDifferences`. Note: Do not manipulate the values prior to calculating the averages.

g. Determine the maximum delay over each week and store this information in the variable called `maxDelays`.

h. Determine the day(s) of the second week on which she arrived to work within 30 minutes. Assign your answer to a `vector` called `daysOfTheWeek`. Note: Only the day(s) of the week should print.

i. Determine the number of days in each week that she arrived within her budgeted time. Call this variable `partI`.

j. On which day(s) did she arrive the fastest in the first week? call your answer `fastDays`. Hint: Use the row names.

k. Subset the matrix of differences to the day(s) of the week on which her commute was similar over the two weeks (ie. both days under budget or both days over budget). Call your answer `similarSub`.

Problem 2

The `Davis` data set in the `car` package contains recorded weight (kg), recorded height (cm), reported weight (kg), reported height (cm), and sex assigned at birth for each subject of a study.

a. Subset the recorded weight and reported weight columns and save them in a data frame named `weight.metric`. Use `head()` to print the first few rows.

b. Convert the weights in `weight.metric` to pounds (1 kg = 2.2 lbs) and name the resulting data frame `weight.imp`. Print the first few rows.

c. Subset the recorded height and reported height columns and save them in a data frame named `height.metric`. Print the first few rows with `head()`.

d. Convert the heights in `height.metric` to inches rounded to one decimal (2.54 cm = 1 inch) and name the resulting data frame `height.imp`. Print the first few rows.

e. Combine the sex assigned at birth information with all of the imperial weight and height information in a data frame named `Davis.imp` with column names `sex`, `rec.weight`, `rep.weight`, `rec.height`, and `rep.height`. Print the first few rows with `head()`.

f. Determine the number of missing values (i.e. `NA`) in each column of the `Davis.imp` data. Call your `vector` of counts `numMissing`.

g. How many rows of the `Davis.imp` data contain a missing value? Note: Do not make assumptions. Call your answer `numBadThings`.

h. Subset the sex assigned at birth of the subjects with a missing value. Call this new variable `mySubset`.

Problem 3

In your astronomy class you have been tasked to create a data set to record the major characteristics of the 8 planets in our solar system in their order from the sun: Mercury (0.39 AU), Venus (0.72 AU), Earth (1 AU), Mars (1.52 AU), Jupiter (5.2 AU), Saturn (9.54 AU), Uranus (19.18 AU), and Neptune (30.06 AU). The first major characteristic is the type of planet: the closest four are `"Terrestrial"` and the farthest four are `"Gas"`. The next major characteristic is the planet's diameter relative to the diameter of Earth: 0.382, 0.949, 1, 0.532, 11.209, 9.449, 4.007, 3.883; followed by the planet's rotation across the sun relative to that of the Earth: 58.64, -243.02, 1, 1.03, 0.41, 0.43, -0.72, 0.67. The next characteristic is whether or not the planet has rings: the closest four do not (i.e. `"No"`) and the furthest four do (i.e. `"Yes"`). The final characteristic is the number of moons: Mercury and Venus have `"None"`, Earth has `"One"`, and the remaining have more than one (i.e. `"TwoPlus"`).

a. Create, save, and print the required data frame using the column labels `name`, `distance`, `type`, `diameter`, `rotation`, `rings`, and `moons` for the columns. The `data.frame` should be called `planet_df`.

b. Subset a data frame that contains any planet(s) with a diameter less than two times that of the Earth. Call your answer `smolPlanets`.

c. Determine the distance away from the sun of any planet(s) that have the opposite rotation across the sun of the Earth. Call your `vector` `leDistances`.

d. Subset the data frame to show only the name, distance from the sun, and type of any planet(s) that have a larger diameter than Earth. Call your `data.frame` `bigPlanets`.

e. Subset the data frame to show only the name and presence of rings of any planet(s) that have more than one moon. Call this variable `manyMoons`.