

Problem set 1

due Monday, September 9, 2024 at 11:59am

Problem 1

Load the `tidyverse` package library into Google Colab.

Problem 1

done

Suppose you track the number of hours you spend studying each day for a week. You store your data in a vector called `study_hours_stringy`. Convert the `study_hours_stringy` vector from strings to doubles using explicit coercion and store it as `study_hours`. Use R's built-in functions to compute the total number of hours you studied that week, the average number of hours studied per day, and the median number of hours studied on a single day. Store the results in a tibble with columns named `Total_Hours`, `Average_Hours`, and `Median_Hours`. Perform a comparison operation on the `study_hours` vector to determine whether each day's study hours were below the median. Finally, use subsetting to extract the days where the number of hours studied was exactly 2 or 4.

Problem 2

Create the matrix given below. Use one of R's built-in functions to append a new row to the matrix made up of all 1s. Multiply every number in the new matrix by 5. Then use subsetting to return the second and fourth row of values in the final matrix.

	[,1]	[,2]	[,3]
[1,]	3	6	9
[2,]	4	7	10
[3,]	5	8	11

Problem 3

Create a data frame that looks like the one below. Return the structure of the dataframe with `str()`. Use subsetting to select all columns that were indicated as doubles. Compute the mean of each column in the subset dataframe using `map_*()` and convert the output to a tibble (use the pipe!).

	age	height	major	score	firstgen
1	30	65	cogsci	100	TRUE
2	45	66	ling	75	FALSE
3	81	72	psych	88	TRUE
4	27	59	ling	97	FALSE

Problem 4

Read the documentation for the [emo](#) package. Install and load the package. Then create a character vector including 8 of your favorite emojis by their keyword (see the docs for a list). Finally, use the package's `emo::ji()` function in conjunction with `map_*()` to print all 8 emojis with one line of code. Combine this output with your keyword vector in a tibble.

Problem 5

Suppose you extend your experiment: you decide to record the number of words your quiet roommate says to you every day for the entire semester. Import the file below from the given url, ensuring that `week` is an integer vector, `words spoken` is a double, and all column names are valid variable names in R. Take care to use the approaches we discussed in lecture to identify and handle any issues. Use `glimpse` to show the resulting tibble. Finally, use these data to recreate (as faithfully as possible) the figure below.

```
"https://kathrynschuler.com/datasets/quiet_roommate.csv"
```

Problem 6

Problem 6 makes use of the `english` dataset in the [languageR](#) package. From the documentation:

This data set gives mean visual lexical decision latencies and word naming latencies to 2284 monomorphemic English nouns and verbs, averaged for old and young subjects, with various predictor variables.

Install and load the `languageR` library. Convert the `english` dataset to a tibble, then use `glimpse` to show the result. Finally, use the `RTlexdec`, `Familiarity`, and `AgeSubjects` columns to recreate (as faithfully as possible) the figure below.