Check-in 4

Please submit as a pdf file on Canvas. Insert answers inside the R Markdown code chunk so that the PDF shows both the code and the output.

Problem 1: (4 pts) For the first part of this question, find the total number of letters in all of the names of the months. To do this, use `{stringr}` functions to first combine all of the month names into a single string (`JanuaryFebruaryMarchAprilMayJuneJulyAugustSeptemberOctoberNovemberDecember`), then output the number of characters in that string.

For the second part of this question, use `{stringr}` functions to extract the first three letters of each month name then combine them into a single string with `-` characters in between each month: `Jan-Feb-Mar-Apr-May-Jun-Jul-Aug-Sep-Oct-Nov-Dec`

Problem 2: (6 pts) Inside the code chunk below, write a single command (using pipes) that completes the following and assigns the output to `dog_data`:

1. Import `stevens_etal_2020_obed_data1.csv` from the following URL: `https://decisionslab.unl.edu/data/stevens_etal_2020_obed_data1.csv`
2. Include the following variables in this order: `class`, `dog_age`, and columns that include the text `personality_`.
3. In the `dog_age` column, replace the text `"< 1 year old"` with `"0.5"`.
4. In the `dog_age` column, replace the text `" years old"` and `" year old"` with no text.
5. Convert the `dog_age` column to a numeric data type.
6. Give a `glimpse()` of the data.

Problem 3: (6 pts) We want to extract semester information from the `class` column. Do the following to the `dog_data` object without assigning to a new or old object.

1. Using a `{stringr}` function, create a new column called `semester` that includes only the first character in `class`.
2. Using a `{forcats}` function, replace the letters F, S, and U in `semester` with Fall, Spring, and Summer.
3. Convert `semester` to a factor.
4. Change the levels of the `semester` factor to be in the order Summer, Fall, Spring
5. Move the `semester` column to be after `class`.
6. Output the structure of the data frame.

Problem 4: (3 pts) There are three dates listed below. Convert each of them into proper ISO 8601 standard format as date data types. Then extract the minute from `pearlharbor`, the year from `dday`, and the weekday (in full words) from `patriotsday`.

```r
pearlharbor <- "7 Dec 41 7:48"
dday <- "June 6, 1944"
patriotsday <- "9/11/01"
```

Problem 5: (5 pts) Write a function called `arithmetic` that takes a numeric vector (`x`) and a single value (`y`) as arguments, along with a `type` of operation to perform on those arguments. The type can be either `add`, `subtract`, `multiply`, or `divide`. The function will then use conditional execution to decide which operation to do then conduct the proper operation (`+`, `-`, `*`, or `/`) on the vector and value. The default `type` should be `add`. Include a message to the console if the type given to the function is not one of the four listed.

Problem 6: (6 pts) Calculate the mean for each of the `dog_data` columns with personality data. Do this three ways. First, use `summarise()` and `across()`. Second, use a `for` loop that iterates through the columns and prints the means. Third, use a `map()` function that outputs a numeric vector. I've included code to convert the tibble to a data frame first, which will help with some of the calculations.

```r
dog_data <- data.frame(dog_data)
```