Check-in 3

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: (7 pt) Inside the code chunk below, write a single command (using pipes) that assigns to the object `dog_data` the following in this order:

1. Imports `stevens_etal_2020_obed_data1.csv` from the following URL:

`https://decisionslab.unl.edu/data/stevens_etal_2020_obed_data1.csv`

2. Includes only the follow variables in this order: `date`, `class`, `dog_sex`, `cgc_test`, all four `cort` columns (using a helper function).
3. Relabels `cgc_test` to `cgc`.
4. Excludes observations where the dog's sex is missing.
5. Creates a new column called `log_cort4` that applies the `log()` function to `cort4`.
6. Sorts the data by `dog_sex` then `cort1`.
7. Gives a `glimpse()` of the data.

Problem 2: (2 pts) Inside the code chunk below, write a single command (using pipes) that calculates the mean, standard deviation, and sample size of the `cort1` variable aggregated by dog sex and CGC status for the `dog_data` object. *Note for this and the remaining questions, do not assign your output to an object.*

Problem 3: (3 pts) Inside the code chunk below, write a single command (using pipes) that does the following to `dog_data`:

1. Creates an `id` variable that is a sequence from 1 to the length of the data frame.
2. Makes the `id` variable the first column.
3. Reshapes the data frame to be tidy such that the four `cort` columns are turned into a column of labels called `time` and a column of values called `cort`.

Problem 4: (2 pts) Inside the code chunk below, write a single command (using pipes) that does the following to `dog_data`:

1. Divides up the `date` variable into `year`, `month`, and `day`.
2. Creates a new column called `session` that combines the year with the class separated by a slash `/` and keeps the `year` and `class` columns.

Problem 5: (2 pts) Looks like more data have come in. After defining `new_data` in the code chunk below, append these new data to the bottom of `dog_data` and sort by *reverse* chronological order by date.

```
new_data <- tibble(date = as.Date("2021-05-18"), class = "U21", dog_sex = "Female",
  cgc = "Pass", cort1 = 0.254, cort2 = NA, cort3 = NA, cort4 = 0.188, log_cort4 = -1.671313)
```

Problem 6: (2 pts) The location data for each class was saved in a different data frame. After defining `location` in the code chunk below, merge the `location` data frame with `dog_data` based on `class` and then move the `site` column after the `class` column.

```
location <- tibble(class = c("U18", "U19", "S19", "F18", "S18", "F19"),
  site = c("south", "south", "campus", "campus", "campus", "campus"))
```