

Check-in 3

Again, the check-ins will review material presented in class but will also require you to think about new concepts, integrate across topics, and search for information. Some will be complex and take time to figure out. Feel free to work in groups on this.

Save this Rmd file in your `checkins` directory. Please submit the assignment as a pdf file on Canvas. Insert answers within the code chunks unless directed otherwise.

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 `mean_cort` that generates a per-observation mean over all of the `cort` columns (ignoring NAs).
6. Sorts the data by `dog_sex` then `cort1`.
7. Gives a `glimpse()` of the data.

Problem 2: (3 pts) For `dog_data`, write a single command (using pipes) that does the following:

1. applies the `log()` function to all of the `cort` columns
2. returns observations only for female dogs who have either passed or failed the CGC (omit missing data).

Note for this and the remaining questions, do not assign your output to an object.

Problem 3: (4 pts) For `dog_data`, write a single command (using pipes) that does the following:

1. returns only dogs who have `cort` values for all four time points
2. creates a new column called `cgc_complete` that converts their CGC outcome to either "Complete" if they passed or failed and "Incomplete" if there is missing data, placed after the `cgc` column
3. creates a new column called `cort_level` that categorizes `mean_cort` as "High" if 0.4 or above, "Medium" if between 0.3 and 0.4, and "Low" if 0.3 or below, placed after `mean_cort`
4. sorts by `mean_cort` from high to low.

Problem 4: (2 pts) For `dog_data`, write a single command (using pipes) that calculates the mean for each of the `cort` columns. *Hint: because `dog_data` is grouped row-wise, you need to `ungroup()` it before anything else.*

Problem 5: (2 pts) For `dog_data`, 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.