

Homework 04: Instructions

Overview

You will be working with the Go/No-Go long data as practice creating variables, some simple and some more complex. Both types of variables will represent the types created for data projects requiring you to create variables based on other variables.

In order to complete the steps, you will need to think about how you would break up goals into sub-goals; in some cases, you will need to use different combinations of relevant functions. I recommend that you use pseudocode, a code syntax-free written description of how to implement the goals. Pseudo code will focus your thinking on the steps and not the code use to implement the steps and will result in a cleaner conceptualization of the problem as well as cleaner code. This pseudo code would become the commented portion of the code using `#`. Once you have your pseudo code, you can use the functions to implement the sub-goals.

The end goal of the Go/No-Go Task that you are working on is to prepare the data for data summaries and/or models. According to the millisecond.com site on which the CHAMP experiment was based, the factorial design is:

```
5 x SOA (100, 200, 300, 400, 500)
2 x Cue (1 = Vertical, 2 = Horizontal)
2 x Target (Go, No-Go)
```

The design is also repeated-measures. However, your liaison may not be interested in your testing this specific model. Instead, they may be interested in comparing performance on the task across other between-subjects groups (e.g., some other factor variable). They may also be interested in predicting individual performance on the task based on other variables, they may be interested in using Go/No-Go performance to predict other outcome variables, or they may be interested in examining other metrics. Whatever the instance may be, you will need to clean up the data and understand the data.

Loading Libraries

Create a code block to load only the libraries you need.

Reading Data

Read the `gng_long.Rds` data file from `/data/gng` and assign the object to `GNG`.

Adding/Modifying Variables

You will now modify the data frame. Rather than creating all variables in one single step, I recommend you ensure that you can create each separately. Once you are satisfied that the variables are created correctly,

combine all of the steps by piping your code. In the end, all of your variables will be created and then assigned to the data frame in order to make the changes permanent. Include only this combined code in the code block following items 1:8.

Simple Variable Creation

1. The `id_wave` variable will be annoying to deal with later and if plots are created with the variable, the name will need to be cleaned up anyway. Create a variable named `wave` that is based on `id_wave`,
2. Move the location of `wave` to be right after `id_wave`
3. Add a variable named `count` that makes all rows contain a value of 1. You

Conditional Variable Creation

You will now create some new variables that will change based on the conditions in the data frame. They will also require you to use other functions to help you create. Various examples were provided in the readings. I will not always provide tips for homework by redirecting you to links in the readings but because some concepts are still new to you, I will. A few functions that you might find helpful with understanding the `ntarget` and `ptarget` variables needed to create your never variable as well as functions that will be helpful when you actually create your variables are provided here:

- `unique()`
- `distinct()`
- `ifelse()` or `case_when()`
- *a more advanced approach:* `str_detect()`
- `startsWith()` from **base R** can be helpful (though not covered in class readings)
- `table()` for cross-tabulation tables

Also, if you ever wish to create a cross-tabulation table for two or more variables to understand how they relate (think Chi-Squared Tests), you can select them from your data frame and pipe them to `table()`. We won't spend much time on this function in class (unless your liaison want you do perform Chi-Squared tests) but you might find it useful to help you understand both `ntarget` and `ptarget`. I'll provide an example here.

```
DATA |>
  select(my_var1, my_var2) |>
  table()
```

4. Add a variable named `cue`.
 - Given the design, the values in the trials should be “Vertical” or “Horizontal”, specifically with the upper case first character because that naming convention may appear better on plots or outputs that might require them.
 - This variable will be based on the `ntarget` or `ptarget` variables in the data frame. There are several ways that you can accomplish this goal using different functions presented in readings thus far; so there are is no single correct approach.
5. Add a variable named `target`.
 - Given the design, the values in the trials should be “Go” or “No-Go”
 - This variable will be based on the `ntarget` or `ptarget` variables in the data frame

6. Add an accuracy variable named `accuracy` based on the response. Consider basing this variable on your `target` variable just created. You will make correct responses 1 and incorrect responses 0, which will later allow you to obtain the mean of the 1s and 0s in order to determine accuracy for each participant.
 - for go trials, responses that are “Correct1” are correct, so make them 1
 - for no-go trials, responses that are “Correct2” are correct, so make them 1
 - responses that are “Incorrect” are incorrect, so make them 0
7. Modify the `rt` variable so that values of 0 are now NA.

Filtering Duplicate Rows

8. Remove rows that contain any NA values for the `trial` variable.

Combining the Full Code

Put your piped code from items 1:8 with all the variables here. Assign the modified data frame to `GNG` so that you change it permanently.

Viewing the Data Frame Contents

Finally, using `GNG`, *remove/select out* the `id_subject`, `id_wave`, `w1age`, and `trial` variables and pipe that data frame subset to the function that you would used to view the top part of the data frame so that I can see the contents.

Finally, knit an `html` file and upload it to: <https://claremontmckenna.app.box.com/f/026ed41483fb4a739f0ddfc7fbf8fd01>