Exercise: Reading, Selecting, & Writing Data Frames

Table of Contents

# 1 **Overview**

For this exercise, you will read data, select variables to subset the data, and write out data files. You will also have to create a seamless workflow that incorporates previous steps already completed so that the data subset is reproducible from beginning to end. Thus, this exercise should support your mental model of breaking tasks into pieces and organizing your code programmatically. These steps will also move you further along on the team project. The steps not completed will become part of the team to-do list.

# 2 **Step 1: Creating A Script File for Task Sub-setting**

Now that you have a script that reads the initial raw data file, cleans it partially, and writes out a cleaned data file, you will create a script that reads in that other script in order to create a subset for the go/no-go task, the symmetry span tasks, or demographic data. If in HW #2 your team identified a survey that was potentially relevant, you would perhaps take the same steps for sub-setting those data as well. I would, however, recommend first discussing this possibility with your liaison.

Team members could collaborate to address a single task or delegate the subcomponents below to separate team members. For the team project, you will need to create at least these three data subsets by writing the code and by writing out the data file. Whatever is not completed in class will become part of the team project to-do list.

1. Create a .R script to subset the cleaned file for a sub-task. If you want to delegate work among team members, choose one and inform your team your plans and add your goal to the weekly work log. The file should be named based on your choice.

Choose one:

* “champ\_all\_waves\_create\_sspan\_subset.R”
* “champ\_all\_waves\_create\_gng\_subset.R”
* “champ\_all\_waves\_create\_demog\_subset.R”

1. Save your script file in /r/<appropriate-sub-task-directory>.

# 3 **Step 2: Basic Editing of the Sub-setting Script File**

1. Inside your script file, add starter code sections for comments and R code as you did for your first raw-data cleaning file. You can find starter helpful code comment [here](https://raw.githubusercontent.com/slicesofdata/fods24/main/r/script_template_block.txt). Realize, however, that your description and relevant sections will differ from file to file but the structure would be the same. You might also consider creating a file as a template script so that content is pre-populated for you every time you need a file.
2. Add a description of the goal of the script.
3. Determine the libraries you will need and in the appropriate section, load those as well as other custom functions.

# 4 **Step 3: Workflow Editing the Sub-setting Script File**

You will now need to ensure that this script is able to perform all of the necessary steps that allow you to subset the data in the event that data are added or changes are made to the data. Rather than pasting the code that reads the raw data, cleans it, and writes out a new data file, will will just want to source the code that performs these steps. In other words, you will need to ensure that this task sub-setting script inherits all of your manipulations with the raw data thus far.

1. Source your “champ\_all\_waves\_initial\_clean.R” script. Remember that this file produces your “champ\_all\_waves\_initial\_clean.Rds” data file.
2. Read the “champ\_all\_waves\_initial\_clean.Rds” data file and assign it to CHAMP.

# 5 **Step 4: Sub-setting the Data Frame**

You will now need to subset the data frame to contain the sub-tasks variables relevant for your task and variables necessary for identifying unique participant data (e.g., their unique id, their school, their wave data) and later merging/joining separate data frames into a combined one using functions like dplyr::full\_join() or dplyr::left\_join().

1. Determine the variable names for unique participant identification.
2. Determine which variables you will need to select for your chosen sub-task.
3. Create a subset data frame with your relevant variables. Make sure not to overwrite your data frame during the testing process.
4. Once you have taken inventory of your variables and you are sure that this subset contains only the variables you need, you can assign the contents to a new object if you like. This assignment step is not necessary if you chain your sub-setting code with the code to export your data (next step) using a piping operator (encouraged).

# 6 **Step 5: Writing out the Subset**

1. Depending on your chosen sub-task export the subset data frame to the respective data directory in /data/<sub-task> and name it one of the following:

* “sspan\_subset\_wide.Rds”
* “gng\_subset\_wide.Rds”
* “demog\_subset\_wide.Rds”

1. Ensure that you can read in the data file.

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