PSYCH 610: HW1

Due Wednesday, September 13, 2023, 1:30pm on Canvas

* Select the Assignments menu and find HW1 on Canvas
* Submit a RMarkdown (.rmd) file and a HTML or PDF file (described below).
* Also submit a word document with Question 7.
* Be sure to change the title above so that “LASTNAME” is your actual last name
* Be sure to edit the name of the .rmd file so that “LASTNAME” is your actual last name
* Make sure you can access the homework folder on Canvas ahead of the due date
* Email the TAs with any questions or trouble submitting the files

## Set up and load packages

### RMarkdown

RMarkdown is intended to combine regular text, code, and output and to create reader-friendly documents. In this course, you will complete your assignment in RMarkdown. You will typically write both your R code and responses to homework questions in an .rmd file and use that .rmd file to “knit” everything together to an output .html or .pdf file.

Sounds confusing? Don’t worry! For this assignment, we’ve provided a template .rmd file you can use (in canvas-homework tab). Make sure you refer to examples in the template .rmd file when reading instructions below. For future assignments, we would expect you to either create your own or adapt and reuse a previous .rmd file.

*External Resources:*

Intro to .rmd: <https://rmarkdown.rstudio.com/articles_intro.html>

.rmd cheatsheet: <https://www.rstudio.com/wp-content/uploads/2015/02/rmarkdown-cheatsheet.pdf>

### Code Chunks and Regular text in .rmd

RMarkdown files are different from R scripts we used for lab. In an .rmd file, what you write directly in the document will be “knitted” as plain text, instead of being treated as R code. You should write your verbal responses to homework questions directly in .rmd files as plain text unless instructed otherwise. You will also need to write and run R code to answer some homework questions. You should do so in “code chunks” in .rmd files. Code chunks begin with three back ticks (```) and “{r}” and end with three more back ticks (see the template .rmd file). You can insert your own code chunks by typing these symbols, or with a keyboard shortcut (command + option + “I” on mac, ctrl + alt + “I” on pc). A code chunk tells R to run the code and print its output in the “knitted” .html file.

### Creating headings in .rmd

You can use hashtags (#) to create headings in your document. Note that this is different in R Markdown than in R scripts we used for lab, where hashtags are used to write comments. Headings will be leveled based on the number of hashtags preceding it. For example:

Headings in .rmd files: Will turn into … in the output .html files:

* # 1st Level Heading **1st Level Heading**
* ## 2nd Level Heading 2nd Level Heading
* ### 3rd Level Heading 3rd Level Heading

Etc…

### Install and Load required packages

Background: For the exercises below (and for all other homework assignments in the future) you will need to load packages. Some packages will need to be loaded for different commands (or "functions"), and other packages need to be loaded for RMarkdown to work properly. We've listed the packages we expect you will need below, but in the future you will need to figure out on your own which packages you may need (note: you may have to install-- not just load!-- some packages).

library(psych) # for describe and describeBy

library(tidyverse) # for read\_csv

library(tinytex) #allows for Rmarkdown to write a pdf document

## Part I: Working with Data in R

For this first assignment, we’re using fake data to practice your new skills.

This hypothetical study is based on work in science and math learning (e.g., Schwartz, D. L., Chase, C. C., Oppezzo, M. A., & Chin, D. B. (2011). Practicing versus inventing with contrasting cases: The effects of telling first on learning and transfer. *Journal of Educational Psychology*, *103(4)*, 759–775.).

In this hypothetical study, suppose participants have been randomly assigned to one of two conditions, Explore or Teach, to learn about a new physics concept.

All participants enrolled in the study completed a pretest, to assess their prior knowledge of the physics concept. Then, participants in the Explore condition attempted to solve related problems on their own to learn about the concept, while participants in the Teach condition received a lesson about the concept directly from an instructor instead.

At the end of the study, participants’ learning was assessed using three posttest measures assessing conceptual understanding (knowledge about the physics concept), procedural competence (ability to work with the concept in laboratory), and perceptual problem encoding (meaningful mental representations of the concept). Thus, the data file includes the following 6 variables:

• sub\_id (Subject ID number)

• pretest (pretest score on knowledge of the physics concept)

• condition (0 = Teach; 1 = Explore)

• post\_con (conceptual understanding score)

• post\_proc (procedural competence score)

• post\_perc (perceptual problem encoding score)

The researchers hope to use these data to answer the following question: do participants’ pretest score and condition (Explore vs. Teach) predict posttest score?

### 1.

Read the data in and check them out.

1. Read in “hw1\_data.csv”, from the course webpage. Name the dataframe “d.”
2. Use one of the functions we learned in lab to inspect the data.
3. As a comment in the code chunk, write out a list of all of the DV(s) and all of the IV(s)/predictor variables present in this dataset. When you are done, you should have 2 lists of variables.

### 2.

Obtain the following basic information about your data:

1. Get a summary of the data.
2. Generate descriptive stats (mean, median, standard deviation, etc.) for all variables in the d dataframe.
3. Get descriptive statistics for just the pretest variable.
4. Generate a histogram for the pretest variable
5. Using one line of code, compute the difference between the posttest conceptual understanding score and perceptual problem encoding score

### 3.

Centering pretest:

1. Create a new variable that is the mean-centered score for pretest knowledge. Give this variable a good name (following the class conventions covered in lab).
2. Check that your new variable is indeed mean-centered using code.

### 4.

Recode the condition variable, which is currently coded as 0 for Teach and 1 for Explore. The new variable should have values -0.5 and 0.5.

### 5.

Create a single global measure of overall learning by averaging the posttest measures:

1. Create a variable that is the average of the three posttest variables (ignore missing data). Name this variable “d$posttest\_m”
2. What are the mean and standard deviation of this mean posttest score? Answer in .rmd as plain text.
3. Generate descriptive statistics of this mean posttest score for each of the experimental conditions.

### 6.

Standardizing post\_con score:

1. Create a standardized variable for the conceptual understanding measure, with an appropriate name (following class conventions).
2. Using one line of code, check your work to make sure this is the standardized score.
3. You have now created several new variables. Write out a command that shows the names of all the variables (also known as “columns”) in the d dataframe (it’s okay if the command shows you more information than just the column names, but finding a command that will give you only this information is a good opportunity to practice Googling!)

### 7.

Create a table of descriptive stats in a Word document (i.e., mean, standard deviation, minimum, maximum) in publication quality format for the following variables: raw pretest score, each of the three raw posttest measures, and the mean posttest score. Give your variables understandable names.

Please see the word doc called pub\_qual\_table on canvas for an example of a publication quality table (Table 1).

Feel free to use this table and edit it for the purpose of this homework. Note: Save this table in a word document as: HW1\_LASTNAME\_TABLE.docx (e.g., HW1\_DONG\_TABLE.docx).

\*Write the code that allowed you to fill in the table.

### 8.

Make a scatter plot with raw pretest score on the x-axis, mean posttest score on the y-axis.

Label the axes and the plot properly, using understandable and precise names.

Consider the plot. What have you noticed? Is there a relationship between pretest and posttest scores? Do the data look fake? Include your answers in .rmd as plain text.

### 9.

Use the command we learned in lab to generate the mean posttest score for each condition.

What do you observe?

Answer in one or two sentences in .rmd as plain text.

## Part II: Reading Questions (Textbook Chapter 1 & 2)

### Answer the following questions in a few sentences directly in your RMarkdown file (not within a code chunk).

### 10. When fitting a model to data, the data analyst faces two conflicting goals. What are those two goals and why are they conflicting?

### 11. What are two consequences of using sum of squared errors rather than the sum of errors as a summary measure of model error?

## Part III: Reflection

Take a minute to reflect on the materials covered in this homework.

### 12. Which parts did you understand pretty well?

### 13. Which parts appear to you the most unclear or difficult?

### 14. How long did it take you to complete this assignment?

### 15. If you have trouble knitting your file, please contact Ben, LiChen, or Nick. Congrats on finishing HW 1!

## Knitting an .html File

A screenshot of a computer

Description automatically generated

When you are all done with the HW, click this. An .html file should be saved after knitting is completed.