DATA IMPORT, FORMATTING, AND *TIDYVERSE*

There are three rules that apply to all projects:

a) Follow instructions *precisely*. If I do not tell you what to write on a particular line, leave it blank.
b) Do not use any functions or approaches to problems that we have not yet learned in this course.
c) All code must be *scalable by sample size* unless specifically noted otherwise. This means your code should work equally well on a dataset with N=10 as N=1000.

You've been handed a weird datafile called *week4.dat*. Here's what you know:

1. N=100; each participant completed the study 4 times with different versions of the stimulus.
2. There are 9 study variables.
3. The 9 variables and their names should ultimately be: case number (*casenum*), participant number (*parnum*), stimulus version (*stimver*), date and time of data collection (*datadate*), and *q1 – q5*.

Part 1 – Set up a new R Studio Project with one R script called week4.R

Part 2 – Data Import and Cleaning

1. **Line 1:** Write a comment that says: **R Studio API Code**
2. **Lines 2-3**: Write code to set the working directory to the directory where your R script is saved
3. **Line 5:** Write a comment that says: **Data Import**
4. **Line 6:** Import functions from the package that will give you access to all core tidy packages
5. **Line 7:** Import the week4 datafile as a data frame called *week4_df*. For this import:
    a. Use a function from the library you just imported.
    b. Do not clean the data; just import.
    c. Name the imported variables according to the data file specifications above as possible; however, name the last column **qs**. Be sure to do this *using import function parameters*. You may need to refer to the documentation for the function you're using.
6. **Line 8:** Display a summary of your data frame the tidy way**,** i.e., don't use summary()
7. **Line 9:** Split **qs** into the five variables they should be according to tidy philosophy. Remember to use the correct variable names as specified above. Do not change their classes yet. Ensure that the values you create do not contain unnecessary whitespace.
8. **Line 10:** Coerce all five of your new variables into a more appropriate class using sapply().
9. **Line 11-15:** Convert all values of 0 (zero) *within these five variables only* into missing values. Extra credit if you can do this using one line of code, but five distinct lines is still a full-credit solution.
10. **Line 16-17:** Convert the datadate column into a class appropriate for datetimes.

Part 3 – Data Analysis

11. **Line 19:** Write a comment that says: **Data Analysis**
12. **Line 20:** Anyone that skipped q2 is not a usable case. Create a new data frame called *q2_over_time_df* that contains each participant number on a single row, with values corresponding to each version of the stimulus as columns (i.e., 5 total variables: participant number, 4 conditions)
13. **Line 21:** Using *q2_over_time_df, d*isplay the proportion of usable cases out of all those collected.

Part 4 – Submission