

Assignment 6

EDH7916 | Spring 2020

Benjamin Skinner

NOTE This assignment needs to be completed by the start of the next class. That means everything pushed to your remote GitHub repo before class starts.

Remember, I encourage you to save your work, commit smaller changes, and push to your remote GitHub repo often rather than wait until the last minute.

For this assignment, you will use a combination of the files we've used so far in class. Be sure to set your data directories at the top of the file (assuming that we're working in the `scripts` subdirectory). Because some of the questions involve reading in the data, you can break our one organizing rule that says to read in data at the top — instead just read in the data as needed for each question

You **do not** need to save the final output as a data file: just having the final result print to the console is fine. For each question, I would like you to try to pipe all the commands together. Throughout, you **should** account for missing values to the best of your ability by dropping them.

Questions

- Using a loop, do the following:
 - Read in each of the individual school test files for Bend Gate and Niagara only. (**HINT** The vertical pipe operator, `|`, means **OR** in regular expression patterns.)
 - Within each iteration of the loop, add a column to the data frame that is called `relative_path` and contains the string relative path to the data file you just read in (*e.g.*, if the file is located at `../data/sch_test/by_school/bend_gate_1980.csv`, then `relative_path == "../data/schools/by_test/bend_gate_1980.csv"` in that row).
 - Bind all the data sets together.
- BONUS:** do the same as before, but use `{purrr} map()` function.
- Read in `hs1s_small.csv` and do the following:
 - Using the user-written function `fix_missing()`, convert missing values in `x1ses` to `NA`.
 - Do the following steps:
 - Subset the full data frame to the first 50 rows and pull out the test scores into a vector using the following code:

```
test_scr <- df %>%  
  filter(row_number() <= 50) %>%  
  pull(x1txmtscor)
```
 - Using a `for()` loop, print out the index of the missing values (when `test_scr` equals `-8`).
 - Repeat the same code, but add an `else()` companion to the initial `if()` statement that prints the value if nonmissing.
 - Add an `else if()` between the initial `if()` and final `else()` in your loop that prints "Flag: low score" if the score is less than 40. Also, change your first `if()` statement to print "Flag: missing value" instead of the index if the value is missing.

3. Write your own function to compare two values and return the higher of the two. It should be called `return_higher()`, take two arguments, and return the higher of two values.

Once you've created it, use it in a `{dplyr}` chain to create a new column in the data frame called `high_expct` that represents the higher of `x1stuedexpct` and `x1paredexpct`. Don't forget to account for missing values!

HINT If stuck on what the inside of your function should look like, go back to the lesson in which we did this already — can you repurpose that code in some way?