SOC 4650/5650: Lab-02 - Data Wrangling

Christopher Prener, Ph.D.

February 5th, 2018

Directions

Using data from the `DataLibrary/CourseData/MO_HYDRO` folder, create a well-formatted notebook that cleans Clean Water Act data for Missouri rivers and streams. Your entire project folder system, including notebook output and results, should be uploaded to GitHub by Monday, February 12th at 4:15pm.

Analysis Development (Review from Lectures 01 and 02)

1. In your course folder system, find the `Labs/Lab-02` subdirectory.
2. Add folders within it called `docs`, `data`, and `results`.
3. Copy and paste the data file `MO_HYDRO_ImpairedRiversStreams.csv` from the course `Data` folder into `Labs/Lab-02/data`.
4. Using RStudio, add an R Project to the `Labs/Lab-02` subdirectory.
5. In RStudio, create a new R Notebook and save it to the `docs` folder you created above. Make sure it is fully set-up following the workflow we have been using.

Part 1: Data Wrangling

6. In the data load code chunk of your notebook, use the `read_csv()` function to load the data stored in `MO_HYDRO_ImpairedRiversStreams.csv`. Make sure it is saved as a tibble.
7. Begin by creating a pipeline that:
   (a) Renames variables to `snake_case` en masse using the `clean_names()` function,
   (b) renames the variable `eventdat` to `date`,
   (c) and rename the variable `county_u_d` to `county`.
8. Next create a missing variable summary using `miss_var_summary()`.

9. Create a duplicate observation report. How many duplicates are there in total? How many actual unique observations are there (i.e. if you removed all of the duplicates but kept a single observation for each unique case)?[1]

10. Check to see if there are duplicates in the `perm_id` variable, which appears like it may uniquely identify observations. Is this the case? If it is not, how many duplicate instances are there? If there are more than twenty, remove this code chunk and its output from your notebook to keep its length short and document in your narrative what your findings were.

11. In a pipeline, make the following two changes:
    (a) Create a subset of observations where `county` is equal to `St. Louis`.
    (b) Then keep only the following variables: `yr`, `wbid`, `water_body`, and `pollutant`, and `source`.
    (c) Assign these changes to a new tibble.

12. In a pipeline, edit the following variables in your St. Louis subset to create a new measure and edit an existing one:
    (a) Edit the `water_body` variable for observations that have the value `Gravois Creek tributary`. Change these values to `Gravois Cr. tributary` so that they match how the word "Creek" is abbreviated in the other observations.
    (b) Then make the a similar change for values `Twomile Creek`.
    (c) Then make the a similar change for values `Watkins Creek tributary`
    (d) Then create a new variable named `ecoli` that is `TRUE` if the `pollutant` is `Escherichia coli (W)` and `FALSE` otherwise.
    (e) Assign these changes back into the existing tibble containing the St. Louis subset.

[1] Look at the `dupe_count` variable that is created in your output. Remember that if you duplicate report is long, it should not be included in your notebook! Just document the results.

Part 2: Reproducible Example

13. Create a reproducible example for question 7 (along with loading the packages required for this example and loading the data) and send it to Brandon and Chris in a direct message on Slack along with a well thought out question about either the plot, the code, or the wider process use to create the plot. The `R` you create does not have to be submitted nor do you have to include the `reprex()` function in your notebook.