

SOC 4650/5650: PS-03 - Dangerous Dams in Missouri

Christopher Prener, Ph.D.

Spring 2019

Directions

Using data from the `M0_HYDRO_Dams.csv` data, create a well-formatted R Markdown document that cleans a data set of all Missouri dams in the U.S. Army Corps of Engineers' National Inventory of Dams. Your entire project folder system, including notebook output and results, should be uploaded to GitHub by Monday, March 11th at 4:15pm.

Analysis Development

The goal of this section is to create a self contained project directory with all of the data, code, map documents, results, and documentation a project needs.

- Clone** the lecture-06 repository if you have not already done so.
- Create a project folder system with all of the necessary components, and drag the lab data from `lecture-07/data/lab-06/` into your RStudio Project's `data/` subdirectory.
- Create a `README.md` text file (File > **New File** > Text File). Add a quick description of your project and outline the key directories and files that are included.
- Create a well-formatted RMarkdown document for your data cleaning efforts.
- Load the `.shp` file containing the dam data as well as the extra data for ground layers (the state boundary and major rivers).
- Re-project each layer using the same CRS code (modifying your object names as needed):

```
data <- st_transform(data, crs = 32615)
```

Part 1: Cleaning the Dam Data

This initial section is focused on cleaning the data on dams prior to mapping.

1. Begin by creating a pipeline that:
 - (a) Renames variables to en masse,¹
 - (b) then renames the variable `fid` to `id`,
 - (c) then renames the variable `offname` (dam name),
 - (d) then renames the variable `owntype` (owner type),
 - (e) then renames the variable `damtype` (dam type),
 - (f) then renames the variable `damht` (height in feet),
 - (g) then renames the variable `nhazard` (hazard rating),
 - (h) and assigns these changes back into the existing tibble.
2. Create a report of missing data across variables - do any variables have missing data?
3. Create a report of missing data across observations - do any variables have missing data?
4. Create and evaluate a duplicate observation report for the entire data frame.
5. Check to see if there are duplicates in the `id` variable, which appears like it may uniquely identify observations. Is this the case?
6. Check to see if there are duplicates in the `idNo` variable, which appears like it may uniquely identify observations. This variable is the National Inventory of Dams Identification Number. Is this the case?
7. Based on your answer to the last two questions, which variable is best to use if we want to uniquely identify observations?
8. In a pipeline, make the following two changes:
 - (a) Create a subset of observations where the dam hazard rating is 1 (the highest danger),
 - (b) then retain only the following variables that were listed under question 2,
 - (c) and assign these changes to a new tibble.
9. In a pipeline, edit the following variables in your high dams subset to create a new measure and edit an existing one:

¹ The following instructions assume you used `"lower_camel"`, so some adaptation may be required.

- (a) Edit the `ownType` variable's values so that they are more descriptive.² The following definitions apply:

F = Federal
 L = Local Government
 P = Private
 S = State
 U = Public Utility

² *Hint:* Make sure you wrap your values in *double quotes*. You will need either five instances of the `mutate()` function combined with `ifelse()` **or** one instance of the `mutate()` function combined with `case_when()`.

- (b) Then make a variable that is `TRUE` if the dam type is RE (an earthen dam), and `FALSE` otherwise.³
- (c) Finally, assign these changes back into the existing tibble containing the high dams subset.

³ *Hint:* Make sure you wrap your values in *double quotes*.

Part 2: Mapping the Dam Data

Create a map of dams in Missouri colored by their height. Use the state boundary and major rivers as ground layers to provide additional context. You should include the other map layout elements necessary, and the map should be exported to your `results/` folder as a `.pdf` at 500 dots per inch. Feel free to use either `ggplot2` or `tmap` for mapping. This will require a modification or two to your mapping approach:

- `ggplot2`:
 - use `mapping = aes(color = variable)` in your `geom_sf` call to shade your points by a particular value instead of the `fill` argument we've used previously
 - use `scale_color_viridis()` or `scale_color_distiller()` instead of the `scale_fill...` functions we've used previously; these are used with the same arguments you used with the `scale_fill...` functions
- `tmap`:
 - use `tm_symbols()` instead of the `tm_polygons()` function we've used previously; `tm_symbols()` can be used with the same arguments you used with `tm_polygons()`

Part 3: Small Multiples of Owner Type

Once we have our overall map, we'll also want to create an additional map that breaks out your data using the new variable owner type variable you created. The map should be exported to your `results/` folder as a `.pdf` at 500 dots per inch. Feel free to use either `ggplot2` or `tmap` for mapping. Use the same code you used in Part 2 to create your map, but add:

- `ggplot2`:
 - add `facet_wrap(~variable)` as another step to your `ggplot()` call and be sure to include a plus sign
- `tmap`:
 - add `tm_facets(by = variable)` as another step to your `tmap` call and be sure to include a plus sign

Part 4: Small Multiples of Dam Type

We also want to create an additional map that breaks out your data using the new variable you created that is `TRUE` if the dam is an earthen one and `FALSE` otherwise. Before you begin mapping, use the following sample code (modified to match your object and variable names) to drop missing observations from your data:

```
data <- filter(data, is.na(earthen) == FALSE)
```

Use the state boundary and major rivers as ground layers to provide additional context. The map should be exported to your `results/` folder as a `.pdf` at 500 dots per inch. Feel free to use either `ggplot2` or `tmap` for mapping, and use the same faceting approach you used in Part 3 (i.e. start with your map from Part 2 and add the appropriate faceting function).