**Folder Structure for LGA Profiling**

The folders contain the following:

**Raw\_to\_Cleaning\_Process**: The script that applies the cleaning log: “LGA\_Profiling\_one\_to\_many\_Cleaning.R”

**Raw\_Data:** The ***raw data file***and***cleaning log*** (same or different excel sheets).

**HQ\_Cleaning**: “LGA\_Profiling\_HQ\_Cleaning.R” and the outputs from the script in the folder:

**updated\_cleaning\_log**: The updated cleaning log output from “LGA\_Profiling\_one\_to\_many\_Cleaning.R”

**updated\_data**: The updated/cleaned dataset output from“LGA\_Profiling\_one\_to\_many\_Cleaning.R”

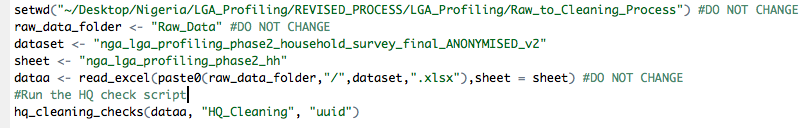
**Analysis**: “LGA\_all\_in\_one.R” and the outputs from that script.

**STEP #1.** Find outliers using “LGA\_Profiling\_HQ\_Cleaning.R”

**FIRST:** Run the entire “FIRST\_LGA\_PROFILE\_FUNCTIONS.R” script to load functions.

**Step 1a.**

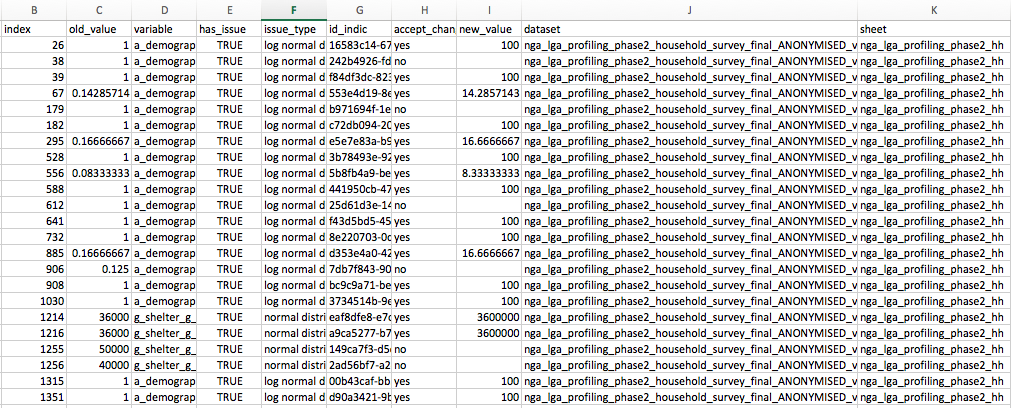
1. Ensure the working directory is set to the “Raw\_to\_Cleaning\_Process” folder.
2. Copy the name of the raw data excel file.
3. Copy the name of the excel sheet where the raw data is located.
4. Check the parametss of the “hq\_cleaning\_checks” function.
5. Run the whole script.



The following error will occur:



However, a .csv file ending in “outlier\_id” is produced. Open this file.

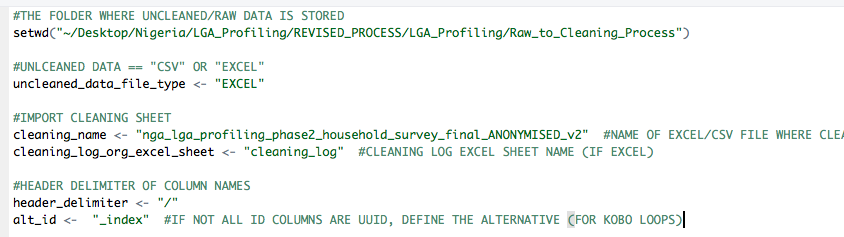


**Step 1b.**

1. Examine the proposed edits and write “yes” or “no” in the “accept\_change” column.
2. Then place the corrected value in the “new\_value” column.
   1. If the entry (survey) must be removed, write NULL (all caps) in the “new\_value” column.
3. Leave the “new\_value” cell blank if the change was not accepted.
4. Save this file as an “.xlsx” file using the exact same name in the “HQ\_Cleaning” folder.
5. Run the script again.
6. Open the “.csv” file ending in “hq\_changes” and copy and paste the contents of that file into the cleaning log.

**Step 2:** Apply the cleaning log to the raw data using “LGA\_Profiling\_one\_to\_many\_Cleaning.R”

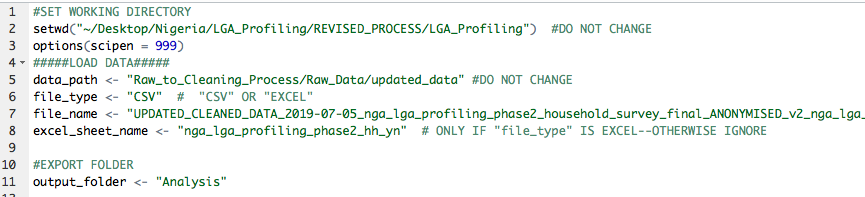
1. Set the working directory to the “Raw\_to\_Cleaning\_Process” folder.
2. Copy the name of the cleaning sheet excel (or .csv) file.
3. Copy the name of the excel sheet where the cleaning log is located.
4. Define how the column names are separated—typically usuing “/”.
5. Define an alternative ID column header, other than “uuid” column.
   1. Example: “\_index” for KoBo loops edits—This is the default and generally should not be changed.
6. Run the whole script and find the updated data in the “updated\_data” folder and the updated cleaning log in the “updated\_cleaning\_log” folder.

****

**Step 3.** Analysis using “LGA\_all\_in\_one.R”

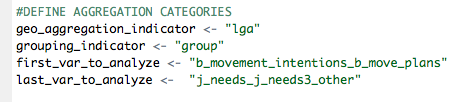
**Step 1a.**

1. Set the working directory to the “root” folder (“LGA\_Profiling”).
2. Ensure the “data\_path” is set to the “updated\_data” folder where the cleaned data output file is located—inside the “Raw\_Data” folder.
3. Ensure the file type is correct (generally “CSV”).
4. Copy the name of the cleaned data file.
5. IF AN EXCEL FILE: Copy the sheet name were the cleaned data is stored.

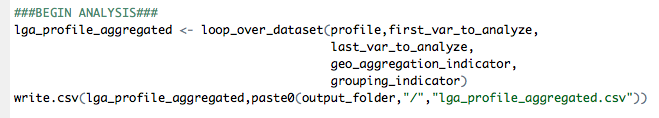
****

**Step 1b.**

1. Ensure the parameters of the “loop\_over\_dataset” function are correctly defined (per the function instructions).
   1. Define the two grouping categories, typically the geographic unit and population group.
   2. Define the first and last indicator to aggregate (leftmost to rightmost).

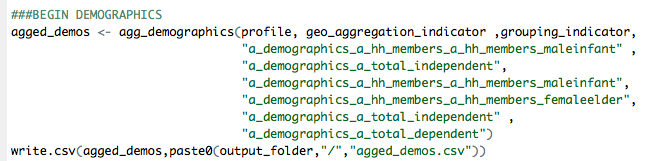


* 1. The function aggregates per-two grouping variables across a dataset from right to left.

****

**Step 1c.**

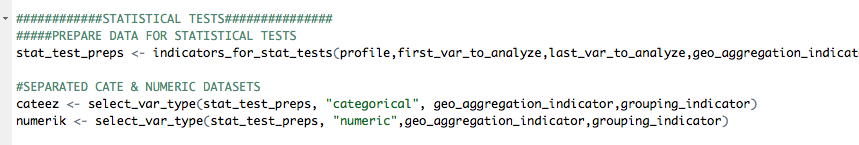
1. Ensure the parameters of the “agg\_demographics” function are correctly defined (per the function instructions).
   1. The function calculates the proportions of each population group per-two grouping variables.

****

**Step 4.** Perform statistical tests on dataset.

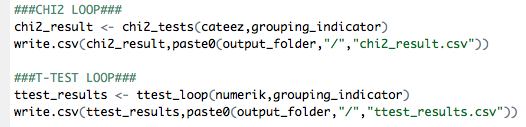
**Step 1a.**

1. Run “indicators\_for\_stat\_tests” function. All parameters are already identified.
   1. This function separates the data according to “numeric” and “categorical”
2. Run the “select\_var\_type” function to subset the list made in step “a” into categorical and numeric indicators.

****

**Step 1b.**

1. Run the “chi2\_tests” function to perform chi2 tests on all categorical indicators, compared between **two (2) groups**. In this case, the “grouping\_indicator”
2. Run the “ttest\_loop” function to perform t-tests on all numeric indicators, compared between **two (2) groups**. In this case, the “grouping\_indicator”

****