Updating AgQuery+

**Definitions and Specifications**

**Folder and File Structure**

The main app directory contains the folders **Data**, **Spatial, Update, and www**. **Data** contains the raw data files, currently stored in csv format. **Spatial** contains the country shapefile needed to render maps. **Update** contains the Excel spreadsheets used to define the app content, and **www** contains additional html and images, like the logic model and the help page. The main folders of interest are **Data** and **Update**.

**Setting Up**

1. **Add Data files (in CSV format) to the Data folder.** 
   1. At minimum, the file should have the columns “hhid” and “province” along with the variables of interest
   2. By current convention, the csv should be named {survey}\_{year}\_{suffix}.csv. The suffix can be user-defined; the survey and year should adhere to the entries in Update/instrument\_list.xlsx
   3. Administrators should pay attention to the size of the data files. While additional variables that may be of interest later can be added to the data files without resulting in corresponding app bloat (because the app will only use the variables listed in the indicators spreadsheet; see below), the full data file must briefly be loaded in when processing data. Large files can result in sluggish behavior or crash under-resourced servers. Spreading the data among smaller, thematically-organized files is preferable.
2. **(For new surveys): Add a weights\_{year}.csv file.** This file should consist of the columns “hhid” and “weight.”
3. **(For new surveys): Add an entry to the instrument\_list.xlsx in Update.** This file only requires updating when a new wave is added. Most of the fields here are implemented in the original AgQuery+ but not in AgQuery+ 50x30.

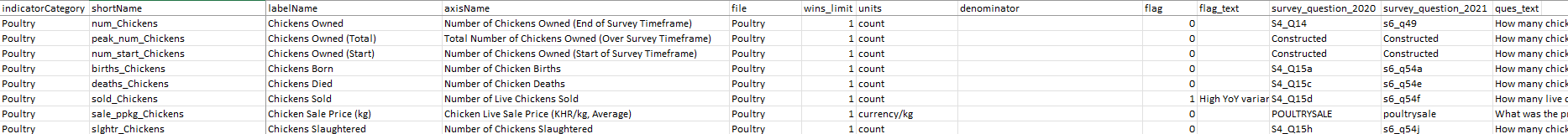


* Survey (String): The survey abbrevation (used to differentiate surveys across countries in AgQuery+)
* Abbr (String): The iso code (used to find data files)
* Symb (Unicode): The unicode emoji characters for the ISO-2 abbreviation. If used in interface elements, this field will render as an icon of the country’s flag. It is not currently implemented.
* Wave (String): the wave abbreviation (not currently used)
* Country (String): The full name of the country. Used to help users navigate across countries in AgQuery+
* Year (Integer): the survey year (for time series graphing and locating data files)
* Yearlabel (String): The full survey period (for category labels)

*The following variables are used for converting currency-based indicators. They are not currently integrated into the app, and the values have not been updated from the model file. This is for illustrative purposes only.*

* Inflation (Double): The coefficient for converting between national currency in the survey year and the national currency in 2017, the current reference year for poverty headcount ratios.
* Gdp\_ppp (Double): The PPP conversion for the survey year based on GDP
* Cons\_ppp (Double): The PPP conversion for the survey year based on private consumption values
* Gdp\_ppp\_2017 (Double): The PPP-GDP conversion for 2017
* Cons\_ppp\_2017 (Double): The PPP-private consumption conversion for 2017

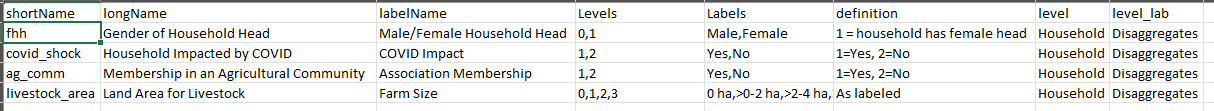
1. **(When adding new or modifying existing surveys): Add the new variables to Update/indicators.xlsx**



* indicatorCategory (String): The policy pathway category to which the indicator is relevant
* shortName (String): Compact, abbreviated name with no spaces used internally to identify the indicator. Matches the variable name in the data file.
* labelName (String): Short but grammatical and title case string used for display purposes, e.g. in the display boxes
* axisName (String): Longer, more descriptive name used for axis titles, may include units
* file (String): The file suffix where the indicator can be found (in the example above, “poultry” is the file suffix because the target data file is CAS\_2020\_poultry.csv)
* wins\_limit (Integer): A single integer representing the percentiles to use for the upper and lower bounds used for Winsorization (upper limit is defined as (100-wins\_limit)/100). Winsorization can be disabled or overidden from within the app using an advanced option (currently disabled) or in the sheet by setting wins\_limit to 0.
* Units (String): A variable indicating whether the unit is countable, monetary, a ratio, a boolean (yes/no) value, or continuous (and type of continuous variable). Used for Winsorization and graphing.
* Denominator (String): For indicators derived based on another indicator (e.g., yield in terms of hectares planted), a shortName representing the denominator of the indicator. Variables with a denominator listed will be weighted by that denominator. Leave blank if none. **Note that this means that the denominator must be included in the same data file, even if it is not used as its own indicator.**
* Flag (Boolean): Indicates whether there is additional information to provide to the user about the indicator
* Flag\_text (String): Additional information required to understand indicator. Implementation is currently in progress.
* Survey\_question\_<Year> (String): The number of the relevant survey question, “Constructed” if the indicator is calculated from another. Used to display information on the data tab; can be left blank. <Year> corresponds to the particular survey year, and a new column should be added for additional surveys, even if the column is left blank.
* Ques\_text (String): The text of the survey question. Can be left blank.

*Note****:*** *removing entries from this Excel sheet will remove them from the interface. It is okay to leave them in the data files, but Policy\_Link.csv must also be updated, or else users will get a notification that data are missing or unavailable.*

1. **(When adding new or modifying existing surveys): add or update the grouping variables in Data.** The grouping variables files contain the variables used to split (or “disaggregate”) indicators for comparison of subgroup means. The files should have household id and household level value of each variable. NAs will be dropped if a variable is used. Updating this file is only necessary if new “groups” variables were added to the datasets.



* shortName (String): As above, matches the name of the variable in the data file
* longName (String): Used for identifying the variable in interface elements
* labelName (String): Used for identifying the variable in graphs/charts; compaction is generally a good idea so the legend doesn’t take up too much space
* Levels (String): Comma-separated list of the numerical or categorical levels of the variable
* Labels (String): Comma-separated list of the labels belonging to each of the levels
* Definition (String): *Not currently used*. In the original version of AgQuery+, this information was displayed in a metadata table to help people interpret downloaded outputs where the variable labels would not carry over to the downloaded file
* Level (String): A category label that can be used to separate variables into separate groups on the interface – in the original version of AgQuery+ this was “Household” and “Within-Household.” It is also used to identify a standalone file (in **Data**) with the variable information to avoid redundancy in the data files.
* Level\_lab (String): The label applied to the group on the user interface

1. **(When adding new or modifying existing policy priorities): Edit Update/Policy\_Pathways.csv**

The policy pathways files (Policy\_Pathways and Policy\_Link, below) are a means to reduce the number of indicators on display at a given time to conserve system resources and avoid overwhelming the user. Policy\_Pathways also governs the data table on the “Pathways” tab*.* For our purposes here, a “pathway” is a potential lever or set of related levers that the government might use to address a headline goal, such as increasing consumption of domestically produced chicken. This table is rendered on the Pathways tab in the app.

A screenshot of a computer

Description automatically generated

* Goal.Id (Integer): A numeric ID used in Policy\_Link to link goals to indicators
* Source.Type (String): The type of publication where the policy pathway was found
* Short.Goal (String): The abbreviated headline goal; it should correspond with indicatorCategory in indicators.xlsx
* Policy.Goal (String): The full headline goal
* Pathway (String): A short description of the policy pathway
* Policy.Instrument (String): A category indicating the type of policy described in Pathway
* Supply.Demand (String): Whether the pathway is supply-side or demand-side

1. **(When adding new or modifying existing policy priorities): Edit Update/Policy\_Link.csv**  
   This is a linking table to connect Goal.Id in the Policy\_Pathways.csv to the indicators in the indicators spreadsheet in order to allow the user to view a simpler subset of indicators that are directly relevant to this policy question. At the moment, its use is optional, and choosing not to update it will not prevent the app from functioning.

A screenshot of a computer

Description automatically generated

* Short.Goal (String): Corresponds to Short.Goal in Policy\_Pathways
* Pathway (String) *Optional*: Corresponds to Pathway in Policy\_Pathways (presented for reference but not used by the app)
* Goal.Id (Integer): Corresponds to Goal.Id in Policy\_Pathways
* Variable (String): Corresponds to shortName in indicator\_list  
  **NB: Goal.Ids are repeated, one per line for each of the variables corresponding to that goal**

Matrix Explanation

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Add for existing indicators data from new survey | Add new indicators from new or existing surveys | Modify which variables are available to users | Modify how users can slice datasets | Modify the policy themes | Modify the instruments under each policy pathway |
| Add/edit data files | X | X |  | X (groups.csv) |  |  |
| Update instrument\_list | X |  |  |  |  |  |
| Update indicators sheet | X | X | X |  |  |  |
| Update grouping\_vars |  |  |  | X |  |  |
| Update Policy\_Pathways |  |  |  |  | X |  |
| Update Policy\_Link |  |  |  |  | X | X |

**Notes for Administrators**

**Publishing updates**: After the data or spreadsheets have been updated, the updates may need to be disseminated (“pushed”) out to other users. For users making modifications to a copy on their own machines, no further steps are necessary; they can run the app from within Rstudio and see the changes immediately. For distribution over Github, someone with repository write permissions will need to upload the edits, and then anyone using that version will need to fetch the updates. For server-hosted distribution, someone with permissions will need to download the new changes to the server from Github, then launch the updated version. For web-hosted (e.g., on the 50x30 web site and not remotely hosted on a server), an administrator will need to update the code and hosted files.

**Localization:** Assuming that the app is running on a system with Khmer language compatibility, for the app to function, the field names in the spreadsheets need to be in English. Everything else can be in Khmer as long as the file names in the **Data** directory match the file names in the related spreadsheet fields. Making the app capable of running entirely on Khmer-language spreadsheets would require modifications to the R code. Khmer file names *are not* compatible with a standard Windows system with English-only localization settings.