How to Update the Data

Andrew, Haowen, Kellie

Summer 2019

This file will walk you through the steps necessary to update the data being used in the Shiny app and elsewhere.

Geocoding the Summer Programs

(Ask Andrew to write this)

Updating the Access Index

(Ask Andrew to update this)

Could... maybe be automated? Though the process for calculating distances from new addresses takes a really long time unless there are very few changes, and with that comes the possibility of weird unforseen errors. I put a readme on how to approach this in the access index folder, but, probably would work best if somebody with some R experience does it manually.

Updating Denver Open Data

Go to DATA/Open_Denver and delete all files. Then go to data_processing/denver_open. Delete the raw_data folder, and open and run the process_open_data_resources.Rmd file in RStudio. This will repopulate the DATA/Open_Denver folder with appropriately processed data.

Note that this updating process relies on the links to individual data sources in Denver Open Data staying the same. If a new data source is uploaded to the website under a different URL, then you will not be pulling data from this source unless you manually change the URL in process_open_data_resources.Rmd. In particular, if new neighborhood level census data is added to Denver Open Data, it will most likely be at a new URL.

If you run into trouble with this step, it probably means that the format of something in the Denver Open Data has changed, which will require troubleshooting.

Updating Google Analytics Data

Go to DATA/google_analytics. Delete the google analytics file, and rerun the CODES/date_processing/Google_Analytics_API markdown. Note that this markdown makes use of a number of string parsing functions defined in the Google_Analytics_Functions R script, as well as the Google Analytics API. To pull data from the API, the user will be prompted to confirm that they have access to the Google account in question through an automatic dialogue box prompted by the initial request. If the user has access, but the API continues to return null results, the adminstrator may have to add/remove the user from the approved list in order to complete the pull. Confirmation is only needed once.

From there, follow the steps in the markdown to parse the path level data retrieved from the API and aggregate this as much as possible to the user level. *Note* at the time of this writing, it does not seem possible to query the API to return data at the level of 1 row = 1 user, so be careful to aggregate to the level of users, and perform all calculations based on the n of users in the data, not the number of rows.

Updating Shiny after all other data is updated

Go to the SHINY folder and run the process_data_once.R file. This does all the preprocessing of the data that's required for the Shiny app, so that the app is quicker to load. You only need to do this preprocessing step once after updating the data sources described above (i.e not before every time you launch the app). It saves all the (updated) data needed for the app in a convenient format, which is then accessed when you open the app.

Note that if you choose to move any of the data sources around (which is highly not recommended), then you will need to update their locations at the beginning of process_data_once.R.

Updating Census Data

This is not necessary for the Shiny app, only for the analysis and report.

Go to DATA/census_raw. Replace those datasets with new and updated raw data according to AmericanFactFinder (AFF) and file names in CODES/Census_update.Rmd. To do this, search for block group level data in Denver county on AFF (advanced search link, put "block group level" in Geographies, put table name(paste from CODES/Census_update.Rmd) in "topic or table name").

Then go to CODES/Census_update.Rmd. File names remain the same, just update ACS year number (eg. the current time point is 2017). Update year number in file paths, then run the rmd file. Final output is Denver_Demographics_BG_XXXX.csv