Caminos de Agua Independence Watershed map maintenance

20th December, 2016

aaron krupp – askrupp@gmail.com

What is the map?

The map shows a point for every water sample *Caminos de Agua* (and perhaps some other groups) has/have ever tested in the *Indepencia* watershed that includes San Miguel de Allende, Dolores Hidalgo, San Luis de la Paz, and many rural communities in Guanajuato, Mexico. Points are color-coded by risk level for various contaminants: fluoride, arsenic, etc. When you select a point, all of the historical data for the site, back as far as 2012, is displayed. When you click these points, or points without historical data, more details are displayed. Some of the points' pop-up info windows have links to water quality monitoring reports done by *Caminos*.

The map is a combination of html, css, sql, javascript, and various javascript libraries (leaflet, carto, and stamen). If that means nothing to you, no worries, but you probably shouldn't me making any functional changes to the map, unless you have significant non-web coding experience or feel like sinking a bunch of time into learning this stuff. That said, it's totally do-able if you've got the time.

Where is the data?

The date lives on the free version of CartoDB, now called carto.com. Since we're using the free version, the data doesn't sync automatically, so we need to re-upload the dataset every time we add more data. For more info, scroll down to the **"how do I upload new data"** section.

Where is the code?

The code lives on GitHub.com. GitHub is a file sharing and editing system the preserves changes and history, allows multiple people to simultaneously work on projects, and, most importantly for us, hosts and runs web code. For editing the code, see **"getting started with code changes"** section below.

How do I upload new data?

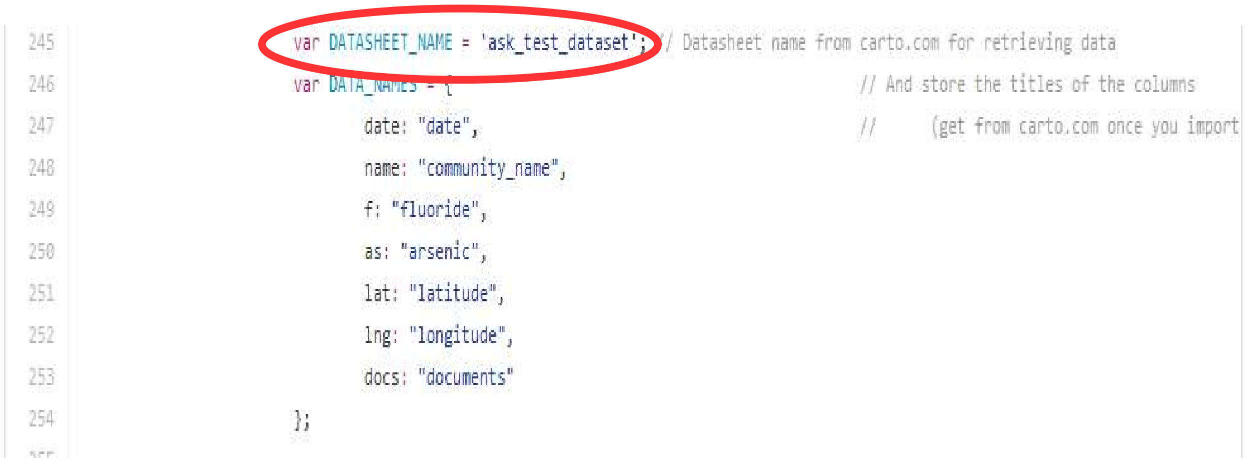
To upload new data

1.Login to the caminos carto account.

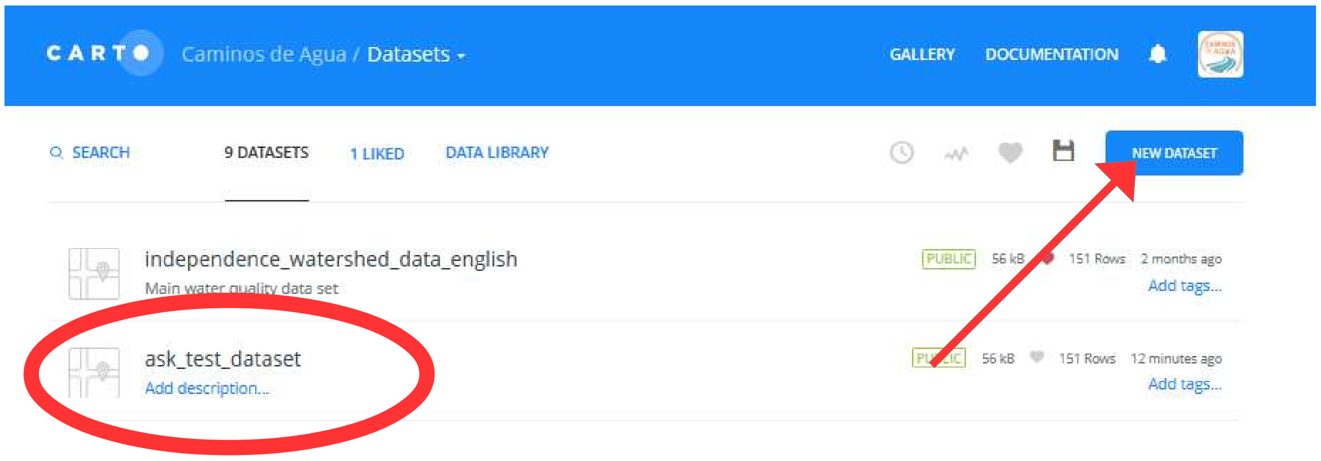
2. Find the dataset that is currently linked to the map. In the index.html file on github, find the line that's like:

```
var DATASHEET_NAME = 'example_datasheet';
```

The name of the carto datasheet is between the quotes. In this case, it is *example_datasheet*.
For the rest of this tutorial, I'll be using *ask_test_datasheet* as in the image below. Notice how the datasheet in carto and the github file have the exact same name. This is important.



```
var DATASHEET_NAME = 'ask_test_dataset'; // Datasheet name from carto.com for retrieving data
var DATA_NAMES = {                       // And store the titles of the columns
        date: "date",                    //     (get from carto.com once you import
        name: "community_name",
        f: "fluoride",
        as: "arsenic",
        lat: "latitude",
        lng: "longitude",
        docs: "documents"
};
```



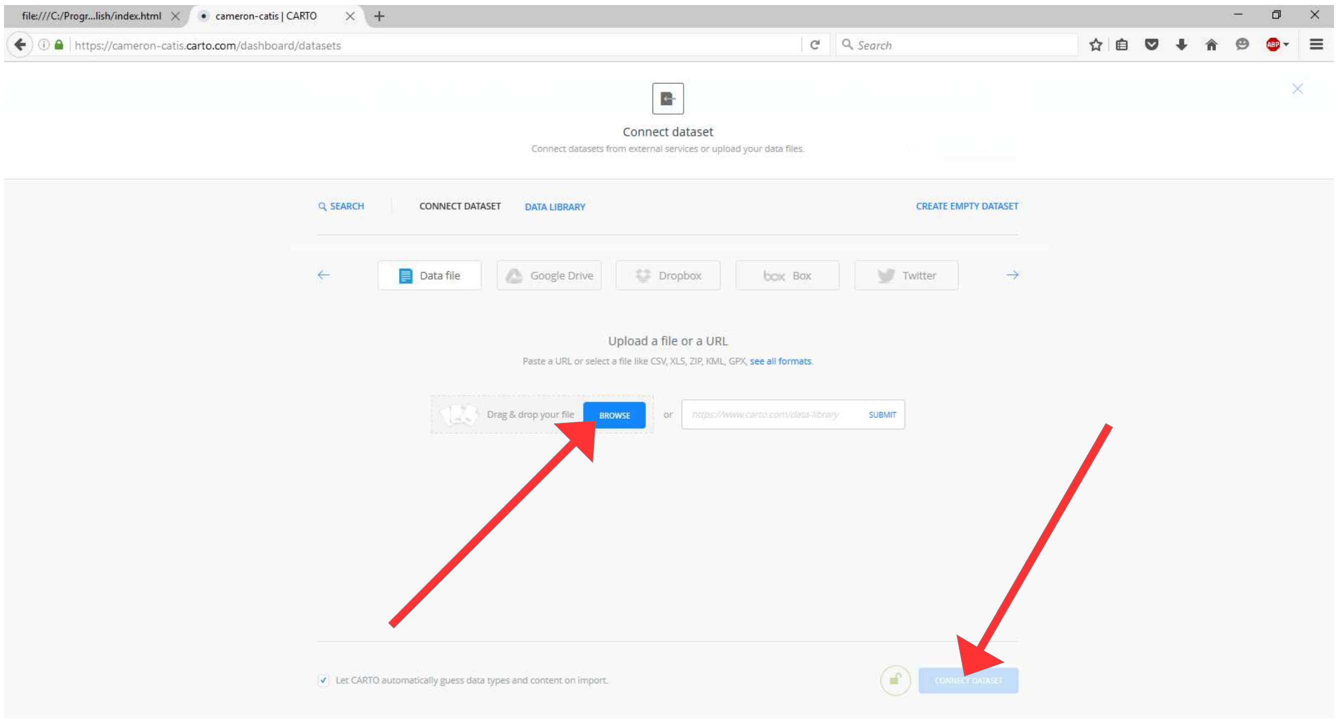
independence_watershed_data_english — Main water quality data set — PUBLIC 56 kB 151 Rows 2 months ago — Add tags...

ask_test_dataset — Add description... — PUBLIC 56 kB 151 Rows 12 minutes ago — Add tags...

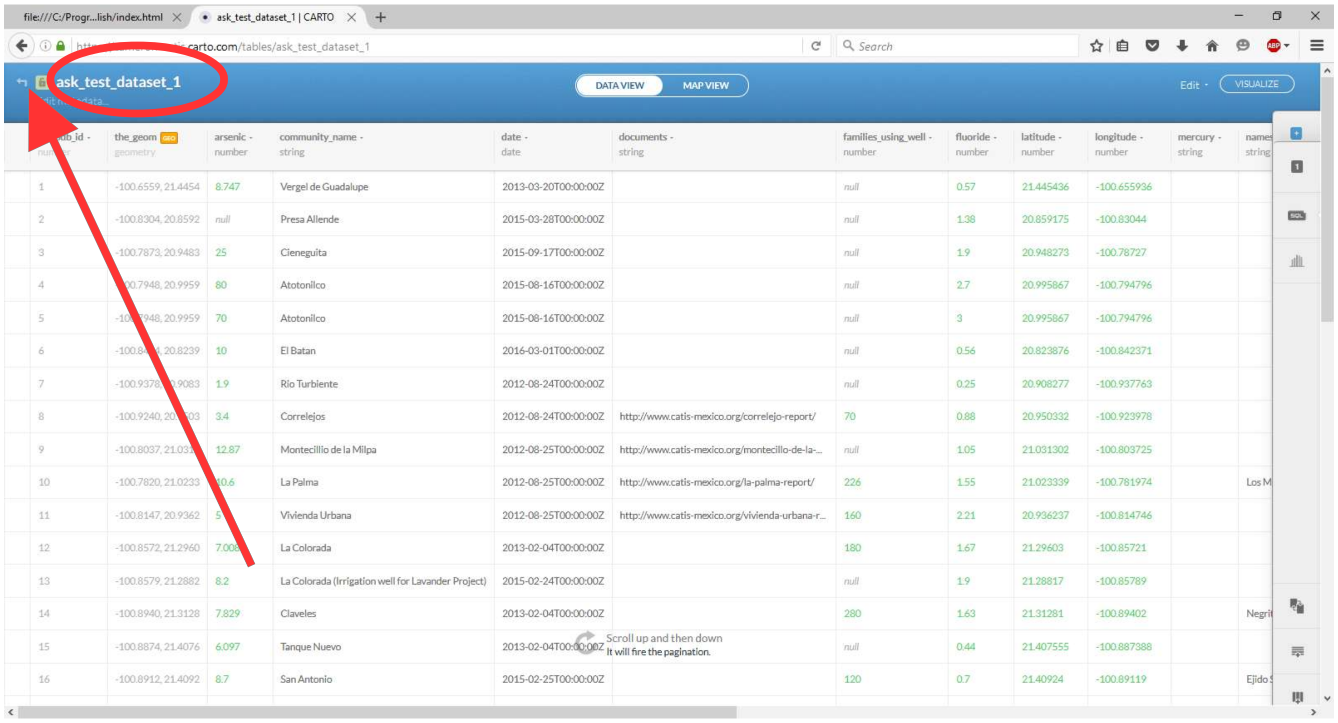
3. Click the NEW DATASET button in the upper right, as in the image above. This opens the dataset upload page, pictured below. Click “Browse,” select the appropriate file.

Note: the file should be in excel format (.xls, .xlsx, etc.), with the column that contains the sample date stored in a date format and separate latitude and longitude columns. Otherwise carto won't be happy.

After you've selected the file, press “Connect Dataset” and you should see a dataset connection box appear in the bottom left.

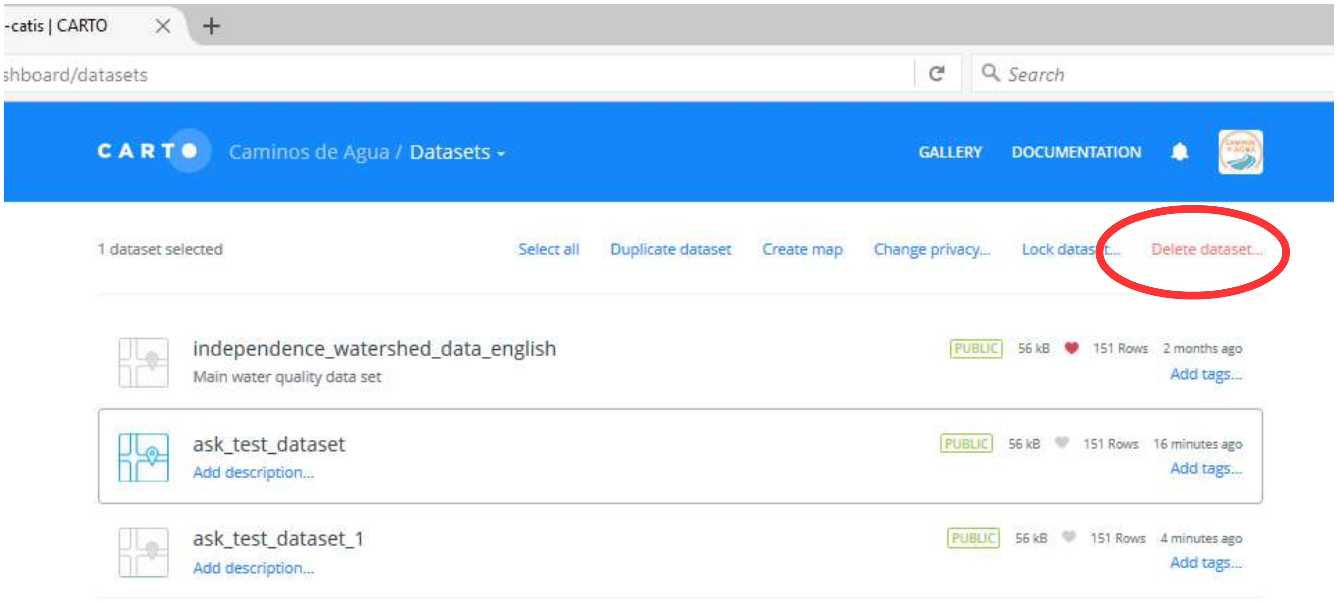

file:///C:/Progr...lish/index.html
cameron-catis | CARTO
https://cameron-catis.carto.com/dashboard/datasets
Search
Connect dataset
Connect datasets from external services or upload your data files.
SEARCH
CONNECT DATASET
DATA LIBRARY
CREATE EMPTY DATASET
Data file
Google Drive
Dropbox
Box
Twitter
Upload a file or a URL
Paste a URL or select a file like CSV, XLS, ZIP, KML, GPX, see all formats.
Drag & drop your file
BROWSE
or
https://www.carto.com/data-library
SUBMIT
Let CARTO automatically guess data types and content on import.
CONNECT DATASET

4. Once the dataset has uploaded, it should automatically open in carto. You'll see that the title, in the upper left has a "_1" stuck onto the end of it, because we already had one with the same name. Press the "go back" button (it's just an arrow pointing left) next to the name. This brings us back to the list of datasets.


file:///C:/Progr...lish/index.html
ask_test_dataset_1 | CARTO
carto.com/tables/ask_test_dataset_1
Search
ask_test_dataset_1
DATA VIEW
MAP VIEW
Edit
VISUALIZE
Scroll up and then down
It will fire the pagination.

[illegible]ub_id number	the_geom geometry	arsenic number	community_name string	date date	documents string	families_using_well number	fluoride number	latitude number	longitude number	mercury string	names string
1	-100.6559, 21.4454	8.747	Vergel de Guadalupe	2013-03-20T00:00:00Z		null	0.57	21.445436	-100.655936		
2	-100.8304, 20.8592	null	Presa Allende	2015-03-28T00:00:00Z		null	1.38	20.859175	-100.83044		
3	-100.7873, 20.9483	25	Cieneguita	2015-09-17T00:00:00Z		null	1.9	20.948273	-100.78727		
4	[illegible]00.7948, 20.9959	80	Atotonilco	2015-08-16T00:00:00Z		null	2.7	20.995867	-100.794796		
5	-10[illegible]7948, 20.9959	70	Atotonilco	2015-08-16T00:00:00Z		null	3	20.995867	-100.794796		
6	-100.84[illegible]4, 20.8239	10	El Batan	2016-03-01T00:00:00Z		null	0.56	20.823876	-100.842371		
7	-100.9378, [illegible]0.9083	1.9	Rio Turbiente	2012-08-24T00:00:00Z		null	0.25	20.908277	-100.937763		
8	-100.9240, 20.[illegible]03	3.4	Correlejos	2012-08-24T00:00:00Z	http://www.catis-mexico.org/correlejo-report/	70	0.88	20.950332	-100.923978		
9	-100.8037, 21.031[illegible]	12.87	Montecillio de la Milpa	2012-08-25T00:00:00Z	http://www.catis-mexico.org/montecillo-de-la-...	null	1.05	21.031302	-100.803725		
10	-100.7820, 21.0233	10.6	La Palma	2012-08-25T00:00:00Z	http://www.catis-mexico.org/la-palma-report/	226	1.55	21.023339	-100.781974		Los M
11	-100.8147, 20.9362	5	Vivienda Urbana	2012-08-25T00:00:00Z	http://www.catis-mexico.org/vivienda-urbana-r...	160	2.21	20.936237	-100.814746		
12	-100.8572, 21.2960	7.008	La Colorada	2013-02-04T00:00:00Z		180	1.67	21.29603	-100.85721		
13	-100.8579, 21.2882	8.2	La Colorada (Irrigation well for Lavander Project)	2015-02-24T00:00:00Z		null	1.9	21.28817	-100.85789		
14	-100.8940, 21.3128	7.829	Claveles	2013-02-04T00:00:00Z		280	1.63	21.31281	-100.89402		Negri
15	-100.8874, 21.4076	6.097	Tanque Nuevo	2013-02-04T00:00:00Z		null	0.44	21.407555	-100.887388		
16	-100.8912, 21.4092	8.7	San Antonio	2015-02-25T00:00:00Z		120	0.7	21.40924	-100.89119		Ejido

5. Click on the **row** of the old dataset, not on the name. That way, it highlights and gives you some options. Once you're 100% sure that your new dataset is uploaded perfectly, delete the old one!



-catis | CARTO

shboard/datasets

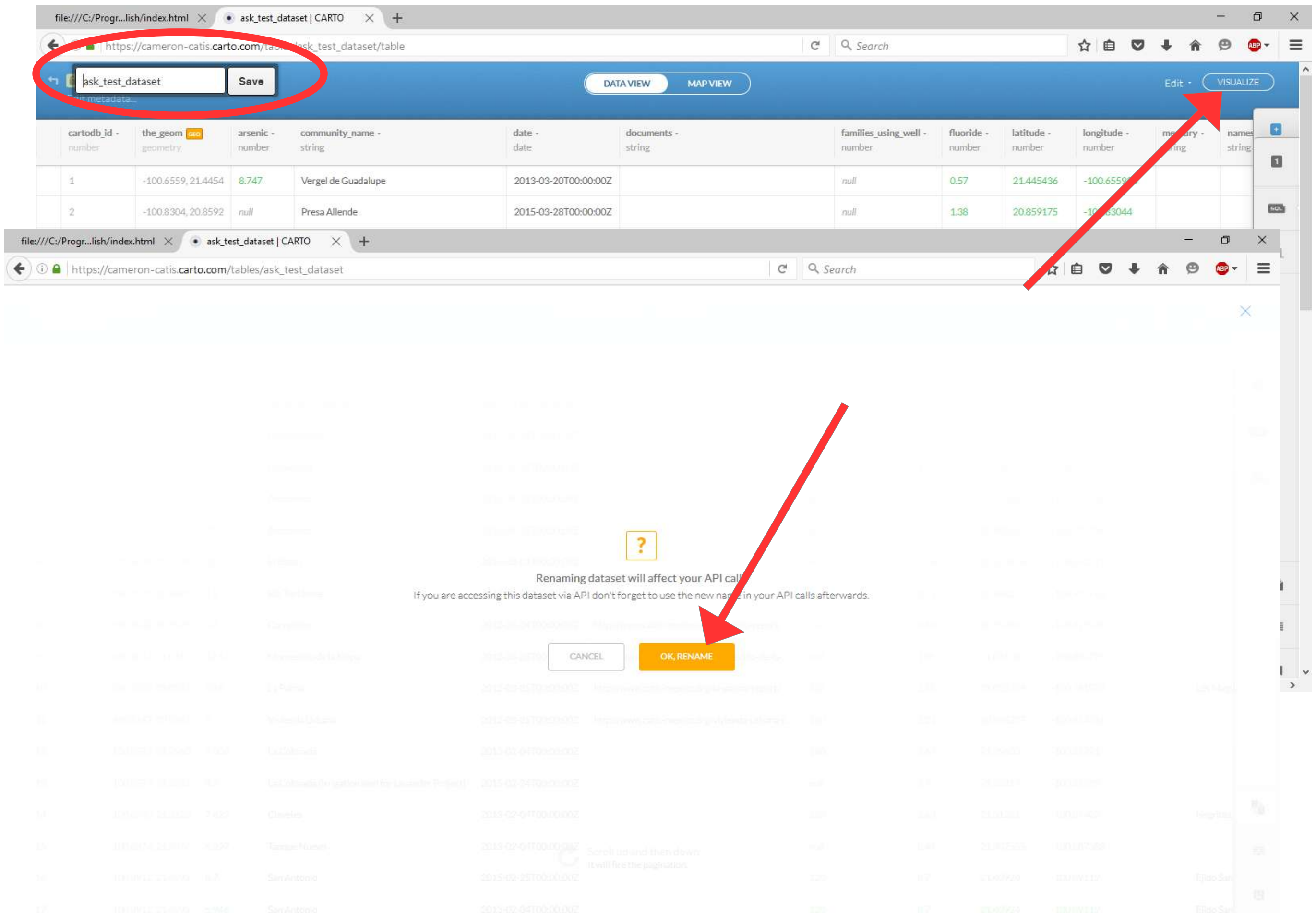
CARTO Caminos de Agua / Datasets

GALLERY DOCUMENTATION

1 dataset selected — Select all — Duplicate dataset — Create map — Change privacy... — Lock datas... — Delete dataset...

independence_watershed_data_english Main water quality data set	PUBLIC	56 kB	151 Rows	2 months ago	Add tags...
ask_test_dataset Add description...	PUBLIC	56 kB	151 Rows	16 minutes ago	Add tags...
ask_test_dataset_1 Add description...	PUBLIC	56 kB	151 Rows	4 minutes ago	Add tags...

6. Then open your new dataset (above, *ask_test_dataset_1*, the "_1" lets you know that it was imported second, as a duplicate name). Click on the name in the upper-left and delete the "_1" or otherwise make sure that its name now matches the name that you found in the code, then push save. After you push save, carto will display an "are you really sure you wanna do that???" screen, as pictured in the second screen below. If you don't see this screen, you've probably done something wrong. Click ok to change. If the change worked, carto will display a message like, "your dataset has been renamed!" or something similar.



file:///C:/Progr...lish/index.html — ask_test_dataset | CARTO

https://cameron-catis.carto.com/ta...ask_test_dataset/table

ask_test_dataset — Save

DATA VIEW — MAP VIEW — Edit — VISUALIZE

cartodb_id number	the_geom geometry	arsenic number	community_name string	date date	documents string	families_using_well number	fluoride number	latitude number	longitude number	me...ry ...ing	names string
1	-100.6559, 21.4454	8.747	Vergel de Guadalupe	2013-03-20T00:00:00Z		null	0.57	21.445436	-100.6559...		
2	-100.8304, 20.8592	null	Presa Allende	2015-03-28T00:00:00Z		null	1.38	20.859175	-10...3044		

file:///C:/Progr...lish/index.html — ask_test_dataset | CARTO

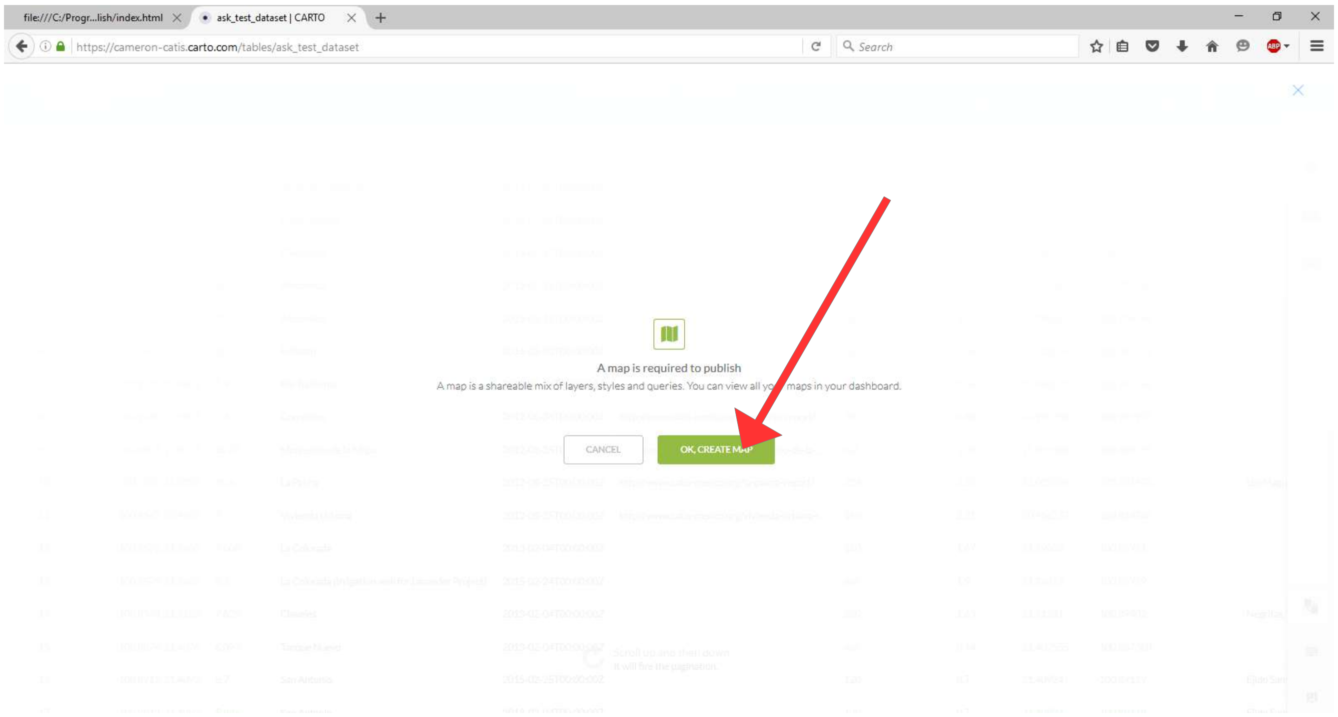
https://cameron-catis.carto.com/tables/ask_test_dataset

Renaming dataset will affect your API calls

If you are accessing this dataset via API don't forget to use the new name in your API calls afterwards.

CANCEL — OK, RENAME

8. To use your datasheet in the map, now, it must be public. To do so, click visualize, as in the image above. This will give you another confirmation screen, as seen below. Again, push the button to accept.


file:///C:/Progr...lish/index.html
ask_test_dataset | CARTO
https://cameron-catis.carto.com/tables/ask_test_dataset
Search
CANCEL
OK, CREATE MAP

A map is required to publish

A map is a shareable mix of layers, styles and queries. You can view all your maps in your dashboard.

Wooohooo! Your data is now named properly on carto.com and will be automatically synched with the map!

Getting started with code changes

The code (that the internet browser uses when you look at the map online) is stored on GitHub. GitHub is a public site where people can store, update, and publish code. Our code is available to the public, but can be modified by invitation only. *Caminos de Agua* has a GitHub account. To change the code, login to our account or make your own account. Once you've made your own account, the *Caminos* account can invite you to modify its repository. (The invitation goes directly to your email.)

Once you've accepted *Caminos'* invitation and have permission to edit the repository (it's called “caminosdeagua/Independence-Watershed-Point-Map-English”) or you're on the *Caminos* GitHub account, you can start making changes to the code!

When GitHub hosts a website, it goes into the “master” branch of the repository and looks for a file called “index.html”. It then creates a site from this HTML file. Find that file and take a look. Whew! Super long. That's cause I suck. My bad. However, there's also lots of grey text. Whew! That's cause I like you. The grey text is comments that describe what's going on in the code. You should be able to read through the comments and have a general understanding of roughly what is happening in each section of code.

[For more help with GitHub's repositories, branches, pushes/pulls or getting GitHub on your personal desktop (the same way as dropbox!) google “GitHub hello-world tutorial.” and play around with your personal account for a while before trying to make any changes to the map].

If if this is confusing so far and/or you don't have experience with web development, check out some tutorials on the structure of an HTML document, javascript, and CSS.

Basically, the document is broken up into chunks:

At the beginning, we bring in a bunch of other scripts/libraries that we'll use to accomplish specific, mapping-related tasks. Then we have a bunch of CSS that describes how lots of the objects on the map are going to look. After the CSS, we have the HTML body, where some objects are defined, like the legends, the dropdown menu, and the caminos logo. The HTML also tells the browser what to do when these objects are clicked on. After the HTML, we have a <script> tag. This is divided into two sections. The top section (in which you'll never see the word, 'function') is code the runs when the website load. This mostly defines visual parameters and lets the map grab some images. The rest of the stuff inside the <script> tag are a set of functions. You probably won't need to mess with these, but if you do, see the full functional descriptions in the comments or in **“how do I make drastic functional changes,”** below.

In GitHub, you can edit the code directly. However, the nice thing about web-development is that you can run your code in any browser, like Chrome, Firefox, etc. So I'd recommend first downloading the code from GitHub (or just copy+pasting it into your favorite text editor – I'd recommend Notepad++, it gives you lots of nice colors – and saving the file). Once you have the code, you can open it in your favorite web browser and, voila! It runs! I like working in Mozilla Firefox, they have a nice debugger and console under Tools > Web Development. Once you're happy that you understand how the code works and how to modify it, you can start making changes in GitHub. Woohoo, best of luck!

How do I get the map onto a website?

Since the code is hosted on github, this one is easy. If you're working on code in a github repository, you can set that repository to create a webpage from its code. (Do so in “settings” once you're in the repository you care about. GitHub takes the file called “index.html” and turns it into a webpage at the link: https://[user].github.io/[repository_name]. So for us, this map is hosted at https://caminosdeagua.github.io/Independence-Watershed-Point-Map-English. If you want to view the map, just enter that link into a browser!

To embed the map in a website, you have a few options. If you're hard-coding a website, you can probably figure it out. Website builders like squarespace, wix, wordpress, etc., often have a box/element that you can copy+paste code into. For embedding this map, you'd probably write a bit of structural code telling the website how to display the map, then put the link.

For squarespace or wix, copy+paste the following into your code box:

```
<iframe width="100%" height="520" frameborder="0" src="https://caminosdeagua.github.io/Independence-Watershed-Point-Map-English/" allowfullscreen webkitallowfullscreen mozallowfullscreen oallowfullscreen msallowfullscreen></iframe>
```

Feel free to adjust the parameters here to size the map appropriately for your site.

How do I make small visual changes?

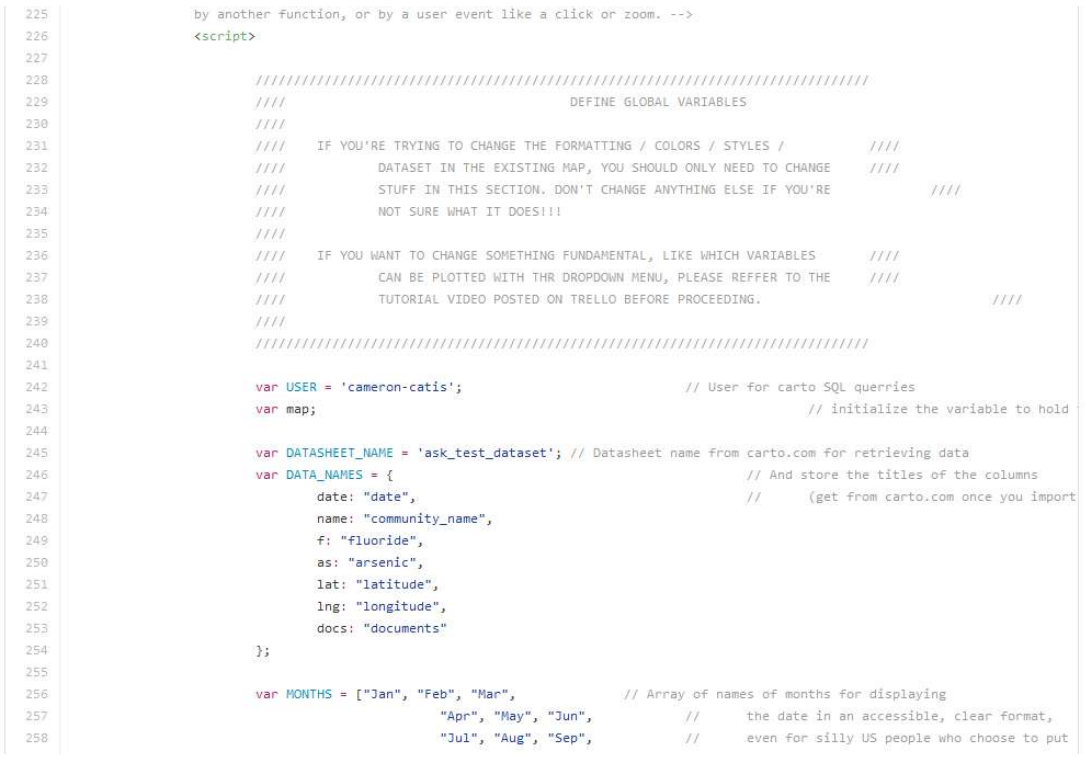
This tutorial assumes you already have a GitHub account (or are using the *Caminos de Agua* one) and have taken at least 10 or 15 minutes to familiarize yourself with GitHub. If you don't know what GitHub is, please go up and read **“Getting started with code changes”** above.

This section explains how to make minor visual changes to the map. These include changing:

- point sizes
- point colors
- the styles/opacity of the points and lines
- the styles of spider x-out button
- the styles of labels and pop-up info windows
- the default zoom and zoom bounds on the map
- the map's base tiles
- the carto attribution in the bottom bar

To make tweaks to the map's current display, login to your (or *Caminos'*) github account. Open the file "index.html" (the main file that holds all the code) in the master repository.

The first section of the <body> <script> is a series of global variables. The top of this section looks like:



```
        by another function, or by a user event like a click or zoom. -->
        <script>

                ////////////////////////////////////////////////////////////////////////////////
                ////                                    DEFINE GLOBAL VARIABLES
                ////
                ////    IF YOU'RE TRYING TO CHANGE THE FORMATTING / COLORS / STYLES /           ////
                ////            DATASET IN THE EXISTING MAP, YOU SHOULD ONLY NEED TO CHANGE     ////
                ////            STUFF IN THIS SECTION. DON'T CHANGE ANYTHING ELSE IF YOU'RE             ////
                ////            NOT SURE WHAT IT DOES!!!
                ////
                ////    IF YOU WANT TO CHANGE SOMETHING FUNDAMENTAL, LIKE WHICH VARIABLES       ////
                ////            CAN BE PLOTTED WITH THR DROPDOWN MENU, PLEASE REFFER TO THE     ////
                ////            TUTORIAL VIDEO POSTED ON TRELLO BEFORE PROCEEDING.                      ////
                ////
                ////////////////////////////////////////////////////////////////////////////////

                var USER = 'cameron-catis';                             // User for carto SQL querries
                var map;                                                        // initialize the variable to hold

                var DATASHEET_NAME = 'ask_test_dataset'; // Datasheet name from carto.com for retrieving data
                var DATA_NAMES = {                                              // And store the titles of the columns
                        date: "date",                                           //      (get from carto.com once you import
                        name: "community_name",
                        f: "fluoride",
                        as: "arsenic",
                        lat: "latitude",
                        lng: "longitude",
                        docs: "documents"
                };

                var MONTHS = ["Jan", "Feb", "Mar",              // Array of names of months for displaying
                                "Apr", "May", "Jun",            //      the date in an accessible, clear format,
                                "Jul", "Aug", "Sep",            //      even for silly US people who choose to put
```

and this section continues for about 150 lines and almost exclusively defines variables. Please download index.html to your computer. There, you can open it in a text editor (I'd recommend downloading Notepad++ at https://notepad-plus-plus.org/https://notepad-plus-plus.org/, but any simple text editor like Notepad that lets you save an .html file is fine). Play with the variables IN THIS SECTION ONLY!!!
For example, changing
>>> var SMALL_ICON_SIZE = [16,16]
to
>>> var SMALL_ICON_SIZE = [60,60]

will make your points enormous.

Or changing
>>> var X_URL = "https://dl.dropboxusercontent.com/s/df3pabfdc7tzr4r/xButton_blue.png";
to
>>> var X_URL = "http://feelgrafix.com/data_images/out/28/992667-che-guevara.jpg";
will replace the x-button that closes the spidered points with a picture of Che Guevara. (You might also want to change the icon size to make that photo larger to adequately appreciate his flowing scruff.

Once you've made a change, save your file and just double click the file on your hard drive to view it in a browser! (Most modern browsers come with debugging tools. In Mozilla Firefox, for example, go to Tools > Web Developer > Web Console to view error messages. These often tell you on what line of your code you made a typo.)

Once your map (now stored on your computer) looks good – and remember, we're only talking about a change or two at a time – make the same edits in the version of *index.html* stored on GitHub. Make sure to test your map before you publish it to make sure you haven't introduced any bugs.

Congratulations, you've changed small visual stuff on the map!

(You can also change some display parameters in the CSS [roughly the 1st 100 lines of the code]). Feel free to play around with those as well in the same way described above.

How do I make drastic functional changes?

This section assumes you already have a GitHub account (or are using the *Caminos de Agua* one) and have taken at least 10 or 15 minutes to familiarize yourself with GitHub. If you don't know what GitHub is, please go up and read **"Getting started with code changes"** above. Although, if you don't have weeks and weeks to sink into this, and aren't already familiar with html, javascript, CSS, and, perhaps, SQL, then making drastic functional changes is probably not worth you taking the time to learn all those things. If you're still comitted to making them, read on! Also, if you are a decent web-developer, please change this code! (I'm not a web person, I know very little about web-dev. This code is structured to be pretty easily modified. May aspect are not optimized. If you're keen to overhaul it and are confident in your ability to do so, be my guest!)

Just note: this document doesn't teach you any of those programming languages, it just describes the structure of the code index.html so you know where to look when you want to make changes. It assumes you're a competent computer programmer and are capable of learning what you need from various documentation sources. If that sounds like you, read on!

If you're still keen on adding or removing some of the map's functions, here is a table of contents of all of the map's functions with brief descriptions:

1. init()
   - Is called when the map first loads. Calls a couple of other initialization functions
   - No return values
2. initMap()
   - Initializes the map object
   - Defines what happens on zoomStart and zoomEnd

- No return values

3. applyBaseMap()
   - Grabs and applies basemap
   - No return values
4. loadAndPlotData(contaminantToShow)
   - Takes in the contaminant to display on the map
   - Reads the JSON data from carto
   - Stores that data in a global var
   - Then calls the functions used to plot points
   - No return values
5. plotMarker(type, data, contam, data_index)
   - Takes in:
     - type --- string, “base” or “preSpider”
     - data --- the full JSON dataset
     - contam --- int/number, the index of the active contaminant
     - data_index --- the index of the releavnt datapoint in 'data'
   - Plots the relevant markes
   - Binds the relevant popups and labels
   - No return values
6. getBin(data, bins)
   - Takes in:
     - data --- a single row of the whole JSON array
     - bins --- a single array from the global array BINS. This can either be full of integers (in the case of arsenic or fluoride, for example, or can begin with a string and continue with the indices of the contaminants to compare
   - Returns a the bin as an integer >= 0.
7. openSpider(data, i, contam)
   - Plot's the spidered points for the clicked-on point
   - No return values
8. window.onclick = function(event)
   - Defines what happens when the window is clicked
   - used to open-close the dropdown menu
   - No return values
9. adjustDDText(contam)
   - Updates the dropdown menu headline text to that of the active contaminant
   - No return values
10. toggleDD()
    - Toggles the dropdown menu open/closed
    - No return values
11. getBasePopup(i)
    - Takes in the index of the relevant row of the JSON data
    - Returns a string to display in the popup infowindow.
12. adjustLatLng(lat, lng, total_pts, i)
    - Takes in:
      - lat --- a latitude
      - lng --- a longitude

        - total_pts --- the total number of points to spider
        - i --- the index of this particular point among the rest of the spidered points
    - Computes the location of the ith spidered point
    - Returns an L.latLng() object with the shifted latitude and longitude.
13. presentIn2dArray(array, value)
    - Takes in:
        - array --- any 2d array WITHOUT REPEATING VALUES
        - value --- a number
    - Checks to see if value is in array.
    - Returns a 2-element array:
        - [0] --- exists --- boolean, true if the value is in the array, false if not
        - [1] --- index --- a 2 element array with the coordinates of 'value' in 'array'
14. getLabel(type, i)
    - Takes in:
        - type --- string "year", "community", or "hist"
        - i --- index of the relevant datapoint in the global JSON data
    - Returns
        - The appropriate label --- string
15. hideLegend()
    - Hides all legeds
    - No return values
16. showLegend(contam)
    - Takes in:
        - contam --- int index of the releavant contaminant
    - Displays the legend for that contaminant
    - No return values
17. closeSpider()
    - Closes the spider that is open (if one is)
    - No return values
18. removePoint(i)
    - Removes the point at index i in the global JSON dataset
    - No return values

There is also a bunch of HTML and CSS code that defines the dropdown menu, *Caminos* logo, and the map legends. Feel free to change those too, their behavior should be relatively self-explanatory.

If you want to change the fundamental behavior of the map, not just the appearance, feel free to modify any of these functions to your heart's content. If you're going to do so, however, please do it responsibly. Use GitHub, save previous working versions, test/debug often, and make sure the version that you're working on isn't publicly displayed on the *Caminos* website. Woohoo, enjoy yourself, and happy coding!!!

More questions?

Please contact aaron at askrupp@gmail.com or through his GitHub account, ask53.