California Stream Gage Dashboards

November 2021

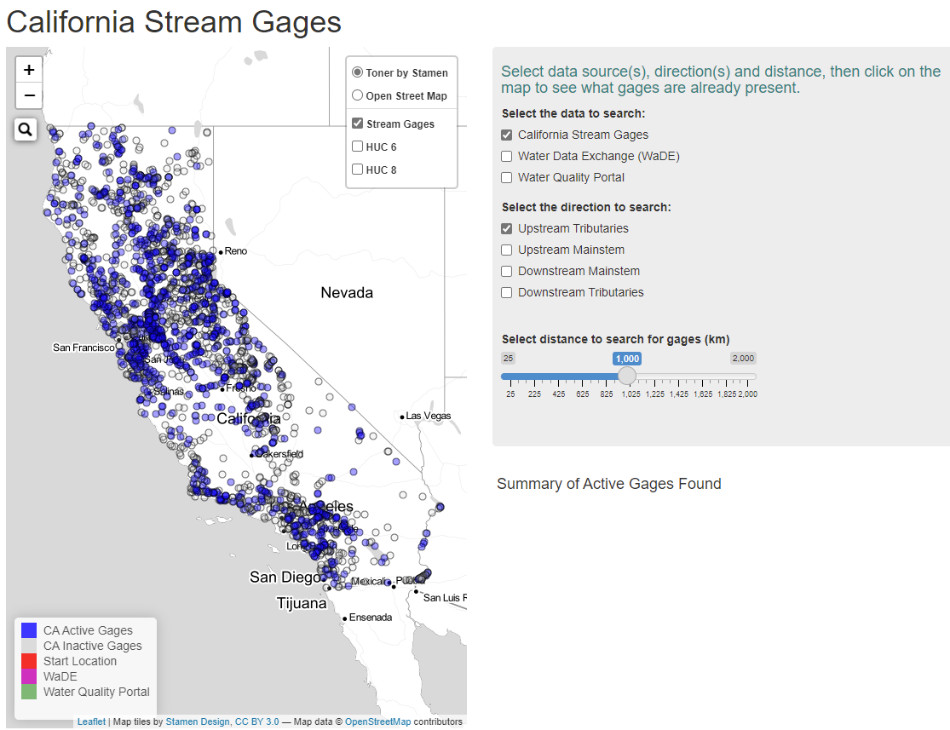
The Internet of Water team developed two prototype dashboards for the stream gauge project. These tools simply show what is possible and can be modified

1. The first dashboard allows the user to click on a location and find all stream gages, water quality measurements, and points of diversion upstream and/or downstream of the clicked location.
2. The second dashboard allows the user to filter the stream gages and toggle map layers that highlight features used to assess gaps in the stream gage network.

# Dashboard 1: Finding Data Along the Stream Network

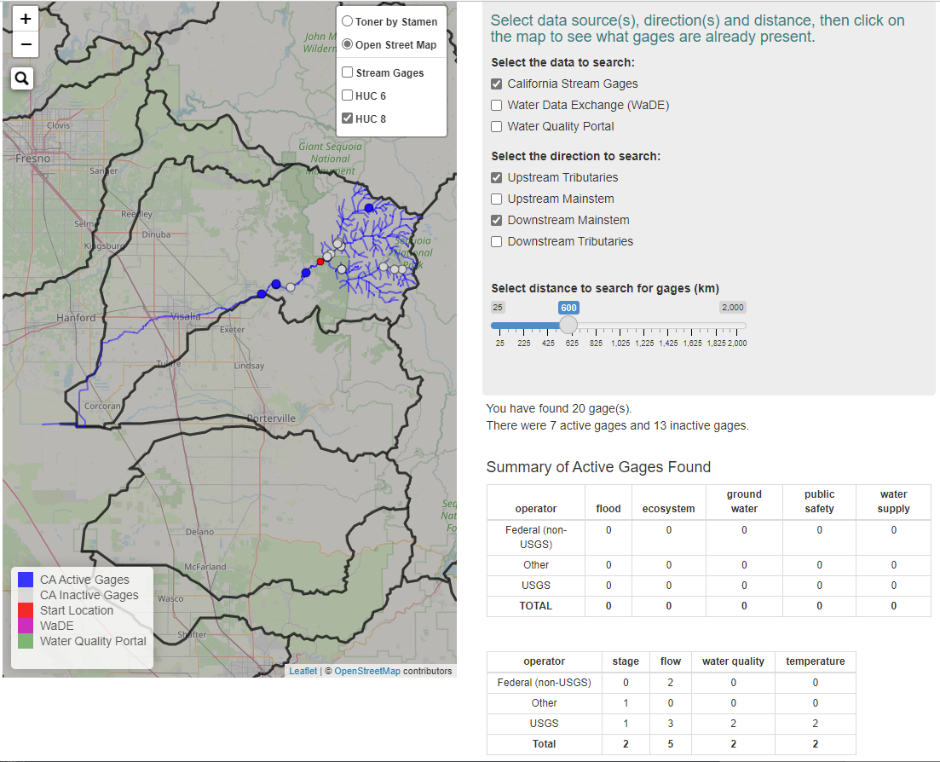
We created an app in R Shiny with the following features:

* Map showing all stream gage locations with active gages in blue and inactive gages in light gray.
  + Upper left corner – zoom in or out. Click on the magnifying glass to type in an address.
  + Upper right corner – toggle the base layer and turn layers on and off
  + Bottom left corner – legend
* Panel in the upper right where the user can make selections:
  + Data to locate (multiple choices can be selected)
  + Direction to trace up and down the stream network (multiple choices can be selected)
  + Distance to search
* Panel in the bottom right showing the results.



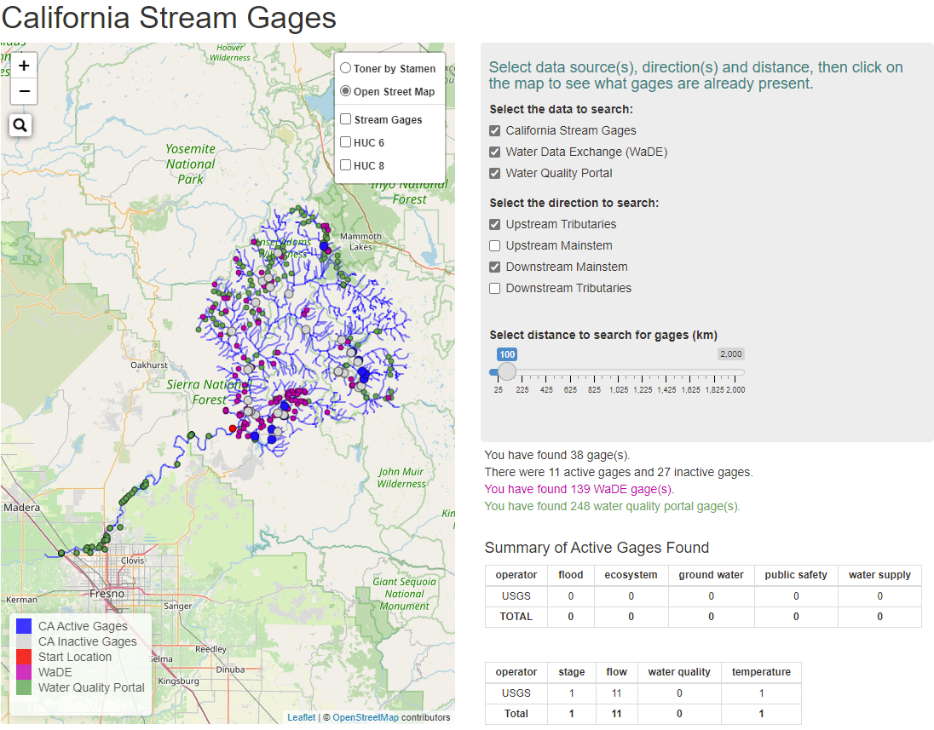
The user makes their selections and clicks on a map location.

In this example, we are searching for ***California Stream Gages*** located along ***upstream tributaries***and ***downstream mainstem*** and within *600* km of the location selected.



The results indicate there were 20 stream gages within 600 km of the selected location (red). Seven of these gages are active. None of them contained information regarding the purpose of the active gages. Most of the gages collected information about flow.

Data may also be found from the Water Quality Portal and the Water and Data Exchange. There are significantly more data held in these portals. This increases the time to search and return results.

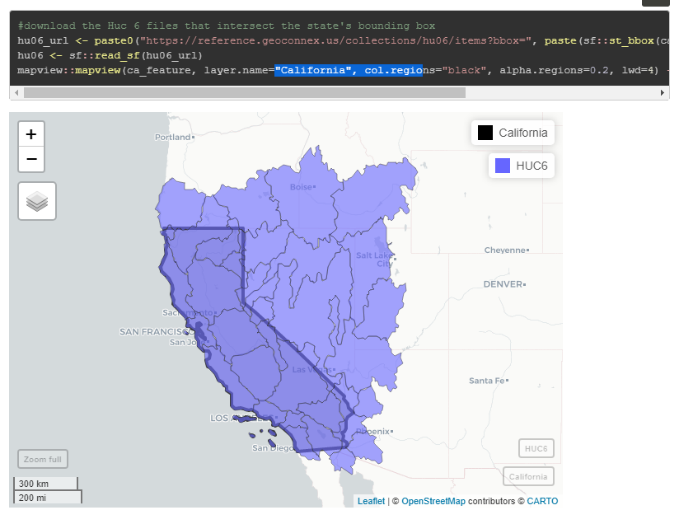


The results here found 139 gages in the WaDE database and 248 measurements in the Water Quality Portal.

## Data Sources

All geojson files were obtained from Geoconnex using R scripts. We created a Rmarkdown file showing how to access these files and clip them to California.

* FILE: *rcode\_inprogress/Markdown\_access\_data.Rmd*



## Shiny App

The rshiny app loads in the gage data and contains functions to trace upstream and downstream based on user selection.

* FILE: *rcode\_in\_progress/ca\_shiny\_app/app\_NLDI.R*

# Dashboard 2: FILTERING GAGES AND OVERLAYS

The second app contains data from the gage gap analysis. The app allows users to do the following:

* Filter stream gages by:
  + status (active / inactive)
  + stream order
  + ownership
  + measurement (e.g. stage, flow, water quality, temperature)
  + management use (e.g. flood, FIRO, ecosystems…)
* Turn overlays on and off
  + HUC 12
  + River basin (HUC 6)
  + Sub-basin (HUC 8)
  + Rivers (displays results from gap analysis using same colors as TNC)
  + Water bodies
  + Stream gages
  + County
  + Urban area
* Change the legend of the HUC 12 to show overlays
  + Conservation status
  + FIRO
  + Conditional Index (not sure what this represents or if cond. = conditional)
  + Threat Index
  + Biodiversity rank (several were available – we chose the overall score here)
  + Percent of poorly gaged headwater, streams, and rivers
    - Could do well gaged, etc.



## Process

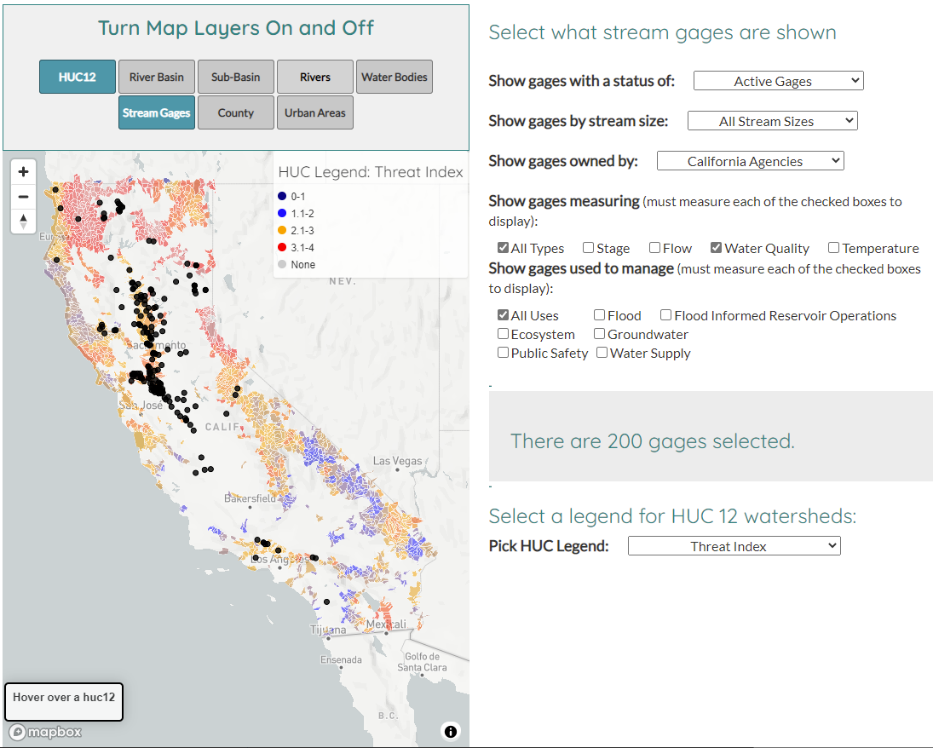
1. Access all the data from geoconnex, nhdplus, and California State Google Drive

* FILE: *rcode\_in\_progress/access\_static\_data.R*
  + This code is fully automated
* FILE: *rcode\_in\_progress/create\_HUCs\_from\_ValData.R*
  + This code is as automated as possible. Some of the downloads were manual. Some of the geodatabase files were too large for my computer to process.

1. Create a single huc 10 and huc 12 file with columns needed for dashboard
   * FILE: *rcode\_in\_progress/create\_HUCs\_from\_ValData.R*
     + The code merging individual shapefiles into a single geojson file for visualization is fully automated.
2. Create a single gage file that includes a column for FIRO gages
   * FILE: *rcode\_in\_progress/create\_HUCs\_from\_ValData.R*
     + The code merging individual shapefiles into a single geojson file for visualization is fully automated.
3. Clean up text in stream gage file to make it easier to explore metadata
   * FILE: *rcode\_in\_progress/create\_HUCs\_from\_ValData.R*
     + The code cleaning up text is fully automated.
4. Create a mapbox id and tiles
   * Several of the geojson files (the rivers, huc12 and waterbodies) are large files that are slow to load. To speed up performance we created a mapbox api and tilesets.
     + Ideally – California will create their unique mapbox api for use
     + To create mapbox tiles you need to download Ubuntu and Tippecanoe.
     + Can access the Mapbox account used by the application
     + Can run tippecanoe. Tippecanoe needs a terminal command line in order to run. This means Windows users will need to follow these steps to install and run an Ubuntu terminal from Windows 10:
     + Open "Turn Windows Features On and Off" from the start menu:
     + Install Ubuntu 18.04 LTS from the Windows Store
     + Restart your computer
     + Open Ubuntu from the start menu:
     + Installing Tippecanoe is often difficult and takes several tries. I found these instructions to be the most helpful for a PC: <https://gist.github.com/ryanbaumann/e5c7d76f6eeb8598e66c5785b677726e>
     + To update mapbox tiles:
     + Open an Ubuntu terminal and navigate to the directory containing your desired GeoJSON files. This can be done using the cd or 'change directory' command. Begin by typing **cd /mnt/c** or **cd ../../mnt/c** to access Ubuntu's mounted C:// drive. Then, use the cd command to navigate to the data directory. You'll know you're in the right directory if typing the command **ls** lists your .geojson files.
     + Run the tippecanoe.sh shell script by typing **./tippecanoe.sh** into the command line. Allow the script to run. If this script ever needs to be changed, simply open it in a text editor and change the commands listed.
     + If the script in the last step executed correctly, you should have new .mbtiles files in your directory, named after the .geojson sources. These files need to be uploaded to mapbox's tile service - Find the tilesets with a matching name in the tiling service website and click its title. From here, click the 'replace' button and drag the matching .mbtiles file into the upload box. The tiles should automatically upload and update for the clients.
     + There is a tippecanoe script that will create the files here:
       1. FILE: *rcode\_inprogress\www\_data\tippecanoe.sh*
5. The html interface is programmed in html (FILE: *index.html*). The css folder contains a styling guide. The scripts folder contains scripts to draw the map, create HUC legends, and interact with the map. This tool was programmed outside of R to take advantage of mapbox tiling services that allowed the large shapefiles (HUC12 and Rivers) to be drawn quickly.

## Map Response

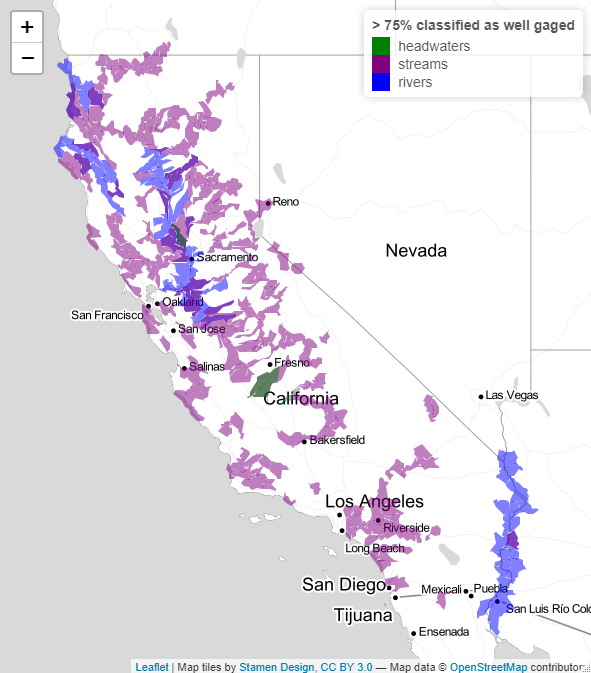
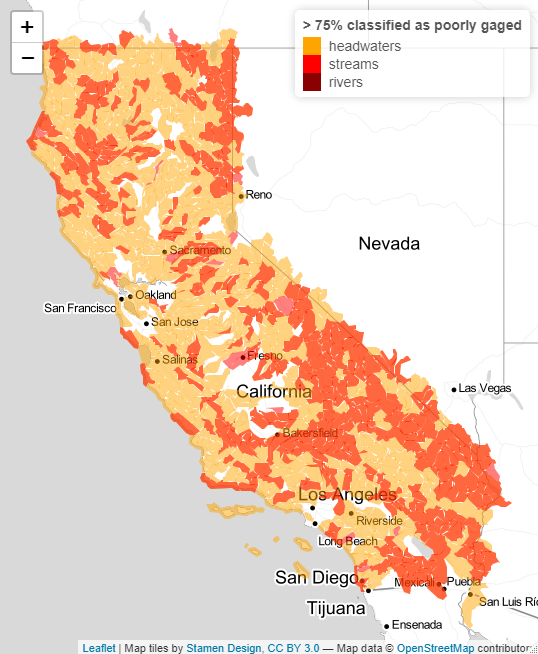
The map does not provide any feedback aside from the number of gages selected. Currently, it only provides the user the ability to filter gages, overlay layers, and change the HUC12 legend.



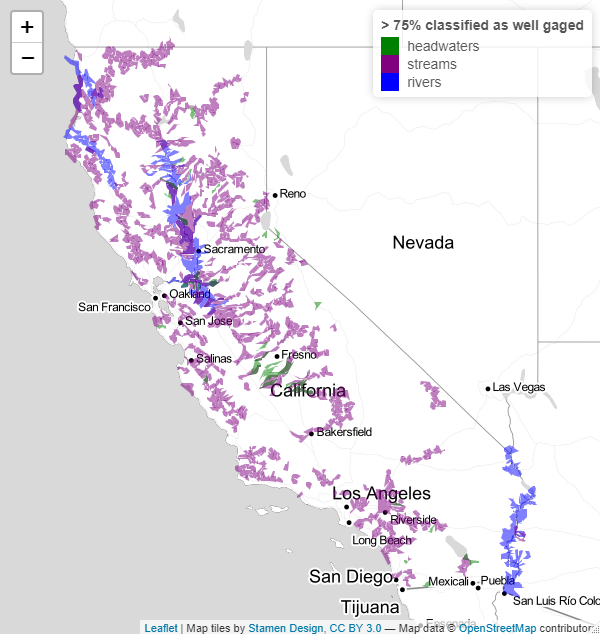
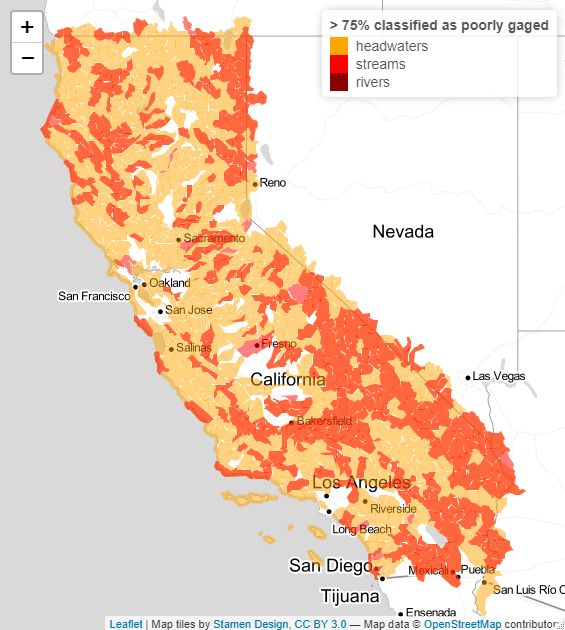
# IDEAS AROUND Visualizations

* Highlight HUC 10 or HUC 12 watersheds that have more than 50% of headwaters, streams, and rivers classified as well gaged
  + Headwaters = stream order 1 to 3
  + Streams = stream order 4 to 6
  + Rivers = stream order 7 to 9
* Tally the river segments classified as poorly, somewhat, and well gaged by classification
* Calculate the percent of each segment classified as well gaged, somewhat gaged, and poorly gaged.

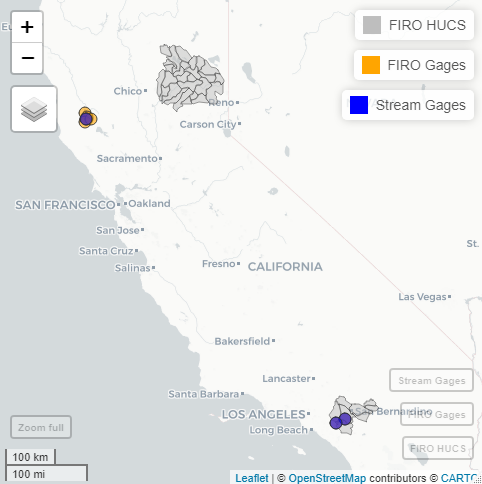
Plot of rivers, streams, and headwaters where more than 75% of those segments classified as well gaged

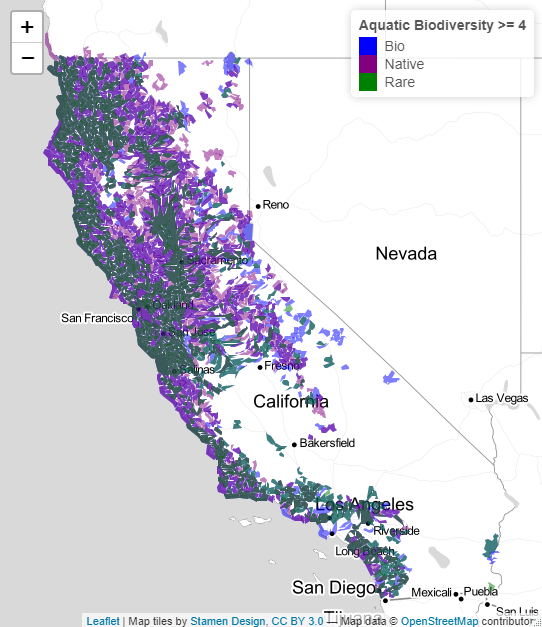
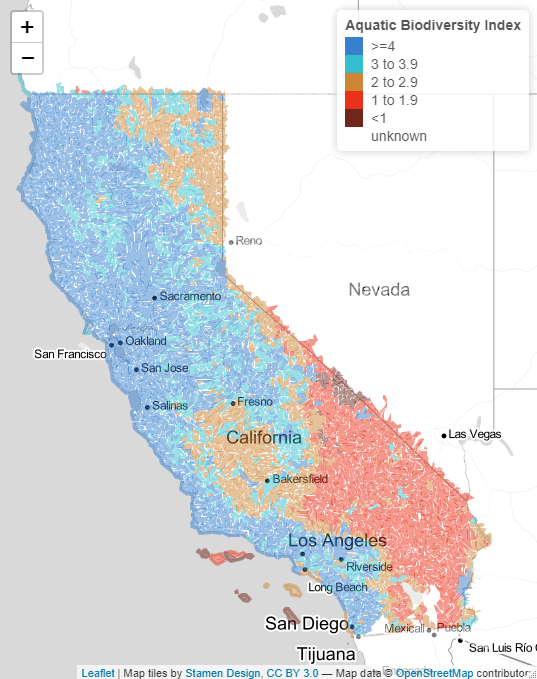
HUC12

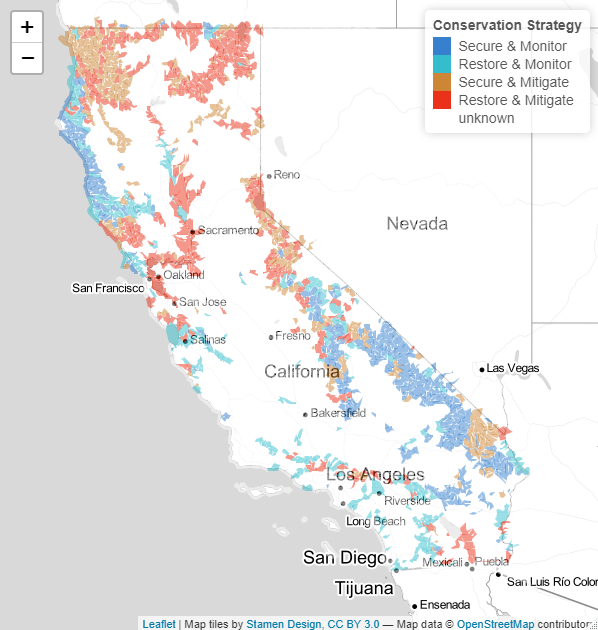
FIRO Gages –and HUCs (note no gages highlighted in the HUCS to the right… not in FIRO gage file. Assumed those gages with FIRO should be classified as floodmgmt and active (all 3 were NA in the original shapefile).



Aquatic Biodiversity – selecting HUCs with the ranking for bio, native, and rare >= 4 (left) and the bio diversity index on right

Conservation Status



The wetlands and natural vegetation files were too large to do anything. I could dissolve the fields but all efforts to simplify broke arcgis, R, and mapshaper.html. California will have to hand a simpler file or calculate percent wetlands in each HUC10 and HUC12. The files as provided would be too large to load on a dashboard.

# Metadata on Gages

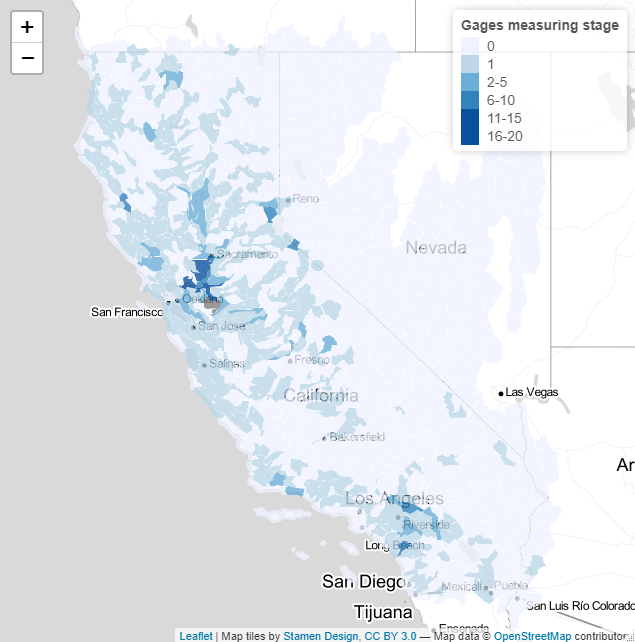
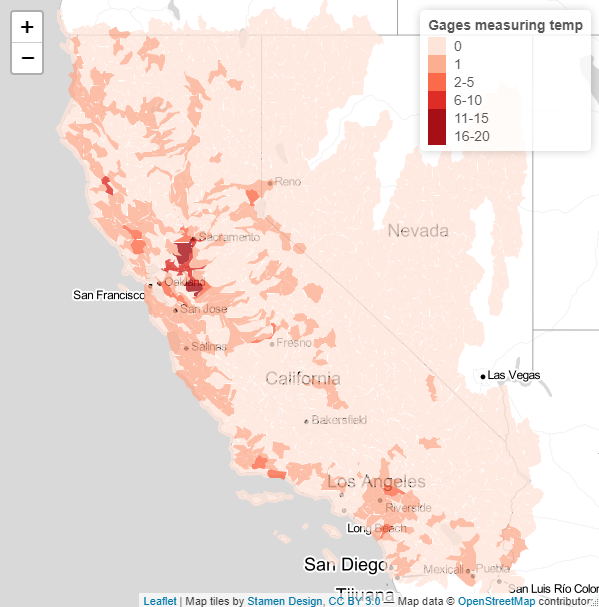
Clean up operator column

|  |  |  |  |
| --- | --- | --- | --- |
| **Operator** | **Includes** | **Active** | **Inactive** |
| California | CA, DWR | 197 | 16 |
| City or County | City, County | 12 | 1 |
| Conservation Districts | Conservation, Consv | 2 | 0 |
| Energy Utilities | Pacific Gas, Power | 31 | 0 |
| Federal (non USGS) | US, Reclamation | 41 | 0 |
| Irrigation Districts | Irrigation, ID | 11 | 0 |
| Other | OTHER, unknown | 109 | 16 |
| USGS | USGS | 670 | 1407 |
| Water Utilities | Utility, Authority, Water | 23 | 0 |

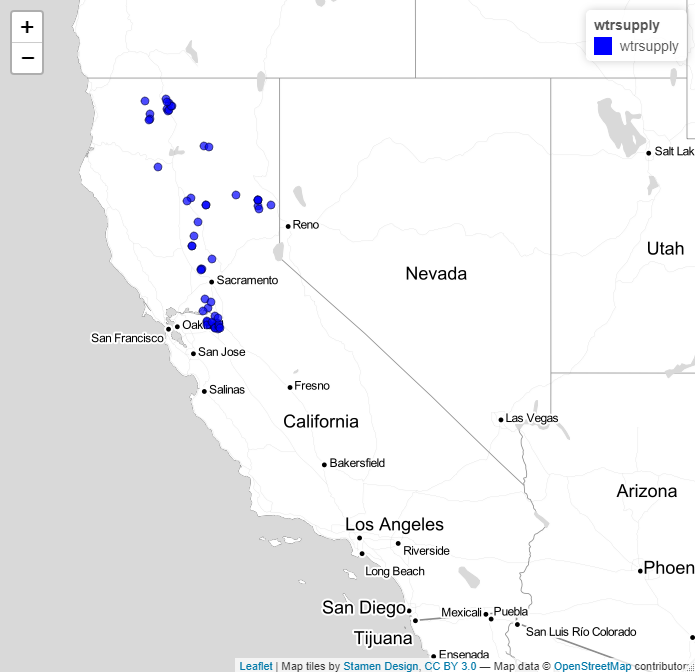
Operator by Stream Type – Active only

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Operator** | **Artificial Path** | **Canal/Ditch** | **Coastline** | **Connector** | **Pipeline** | **Stream - intermittent** | **Stream - Perennial** |
| California | 54 | 27 | 2 | 3 | 1 | 23 | 87 |
| City or County | 2 | 0 | 0 | 0 | 0 | 0 | 10 |
| Conservation Districts | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| Energy Utilities | 1 | 1 | 0 | 1 | 0 | 0 | 28 |
| Federal (non USGS) | 15 | 1 | 0 | 1 | 0 | 6 | 18 |
| Irrigation Districts | 0 | 1 | 0 | 0 | 0 | 2 | 8 |
| Other | 5 | 3 | 0 | 0 | 1 | 25 | 75 |
| USGS | 88 | 41 | 0 | 4 | 6 | 142 | 385 |
| Water Utilities | 1 | 3 | 0 | 0 | 1 | 7 | 10 |

Other options are to calculate the number of gages measuring different things and highlight HUC accordingly. Lots of gages have missing data on measurement and management.



Make circle size based on the number of purposes a gage is fulfilling and highlight certain purposes



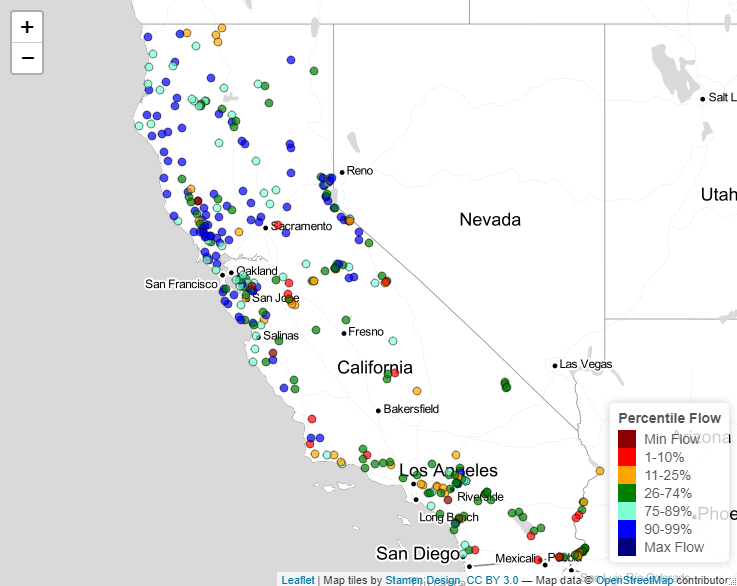
When select a gage we can create a table showing it’s purpose… or when a huc is selected it grabs all the gages

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Site** | **Flood** | **Ecosystem** | **Groundwater** | **Public Safety** | **Water Supply** |
| ANH | Yes | No | No | Yes | No |

We can say how much data has been collected for an individual gage

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute | Stage | Flow | Water Quality | Temperature |
| Collected (Y/N)? | Yes | No | Yes | Yes |
| Days of Data | 13,514 | 0 | 8,752 | 9,496 |
| Years of Data | 37 | 0 | 24 | 26 |
| Real Time (Y/N)? | Yes | No | Yes | Yes |

With Geoconnex and APIs it is also possible to load in live data.

For example – in the script: “pull\_realtime\_streamflow” we can show 7 day average streamflow for CA using USGS data. 

Can pull metadata directly from CA or a selected gage using geoconnex. For example, a user could select on gage CCK and it takes them to this weblink: <https://cdec.water.ca.gov/dynamicapp/staMeta?station_id=CCK>

# Notes

* Geojson files are still large even after simplification. This makes it very slow to load in leaflet (app.R). About 1 minute.
* Mapboxer is a new R package that uses mapboxGL – this greatly increases the load time but still take 5 to 10 seconds on each load and 2-3 seconds for each update. Reading in mapbox vector tilesets does not appear to be an option yet.
* Fastest speeds and best performance are in javascript using mapbox tilesets.