

## SOFTWARE METAPAPER

# HydroCloud: A Web Application for Exploring Stream Gage Data

Martin C. Roberge<sup>1</sup>, Michael P. McGuire<sup>2</sup> and Jie Lian<sup>2</sup>

<sup>1</sup> Towson University, Department of Geography and Environmental Planning, US

<sup>2</sup> Towson University, Department of Computer and Information Sciences, US

Corresponding author: Martin C. Roberge ([mroberge@towson.edu](mailto:mroberge@towson.edu))

HydroCloud ([hydrocloud.org](http://hydrocloud.org)) is a mobile-friendly web application for visually browsing hydrology data from multiple sources. Data providers such as the US Geological Survey (USGS) and the German *Wasserstraßen- und Schifffahrtsverwaltung des Bundes* (WSV) currently serve stream discharge data from more than 10,000 stream gages around the world. HydroCloud allows users to plot these data while out in the field, while also providing contextual information such as the current NEXRAD weather imagery or descriptive information about the stream gage and its watershed. Additional features include a chat mechanism for contacting developers, and the use of local storage for saving data.

**Keywords:** Data exploration; hydrology; stream discharge; web application

**Funding Statement:** This project was supported in part by a grant from the Towson University School of Emerging Technology.

## (1) Overview

### Introduction

The US Geological Survey maintains and distributes a vast amount of hydrologic data for more than 1.6 million stream and well sites; over 16,000 of these sites collect observations between every 5 minutes to hourly [1]. An even greater number of stream gages around the world are monitored by government agencies in Germany [2], the U.K. [3], and Australia [4], among others. These data are provided to the public through RESTful web services in easily parsed formats such as JSON, XML, and the WaterML standard [5, 6]. Unfortunately, the software needed to view these data consist either of desktop applications aimed at a scientific audience [7] or through individual web pages maintained by each agency.

HydroCloud is a single-page web application (<http://hydrocloud.org>) that allows the user to browse stream gage data from a variety of sources directly from the original data provider. It includes a map interface for selecting stream gage sites, a graphing interface for displaying and analysing the stream discharge data, and a summary table view that lists non-spatial data for each site. An additional help page explains how to use the application.

The goal of the HydroCloud application is to demonstrate that it is possible to interact with multiple large scientific datasets in a way that is easily accessible and understood by the general public, while still maintaining the integrity of the data for scientific analysis. To achieve the accessibility goal, HydroCloud features a responsive,

mobile device-friendly design and a clean, simple interface that requires no special understanding of the dataset or application to use. Its use of web technologies ensures that it is viewable on any operating system through a web browser, and requires no special downloads or instructions to install. Because the data are loaded and stored locally, it is theoretically possible to compute any reasonable statistic or analysis on these data within the web app or by a separate software application.

The HydroCloud application is part of the larger HydroCloud project [8, 9, 10], which explored the possibilities of integrating different hydrologic datasets in an Online Analytical Processing (OLAP) cloud environment.

### Implementation and architecture

HydroCloud was written entirely using HTML 5 technologies (JavaScript, CSS, HTML) to ensure that it can be used on any operating system using any modern, standards-compliant web browser, including Chrome, Safari, Firefox, Opera, MS Edge, and IE10. The application integrates a variety of data services in one tool, while aiming towards a goal of interoperability and responsiveness.

The heart of the HydroCloud application is the framework, which mediates between the various data sources (the Model) and the GUIInterface (the View) through the ViewModel. For this application, we used Knockout.js (<http://knockoutjs.com/>), a smaller framework that utilizes two-way data-binding to link changes between the View, ViewModel, and Model. For example, the framework

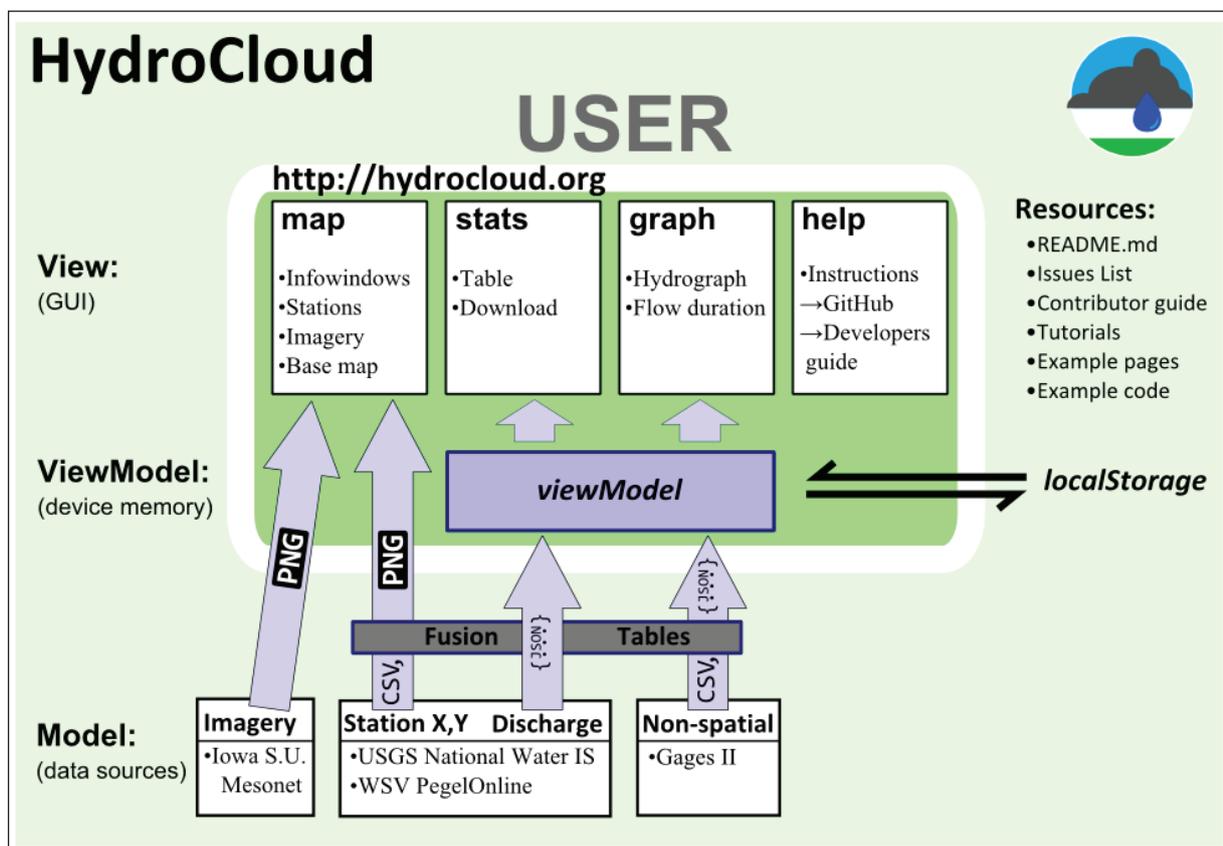
might update what is seen in a graph (the View) in response to new data that was just added to the ViewModel from a pending query. Alternatively, Knockout.js might update the ViewModel in response to a user's interactions with the GUI aspect of the View. In HydroCloud, the ViewModel is named *viewModel*, and includes properties that record the state of the GUI, such as whether the legend is visible or not, the type of graph that the user has selected, and whether the map, graph, table, or help view should be showing. The viewModel object also stores a list of sites that the user has selected, as well as the associated data for those sites.

The styling of the page and associated CSS is handled using Twitter Bootstrap. Bootstrap controls the functioning of the pull-down menus, the display of the legend box, the positioning of the fixed headers and footers, as well as the responsive design, which allows the page to display differently for different sized screens. Bootstrap's carousel feature has been modified slightly from the original to allow the switching between the map, graph, stats, and help views. Icons are used to highlight the state of the application, as well as to allow users to switch between views. jQuery is used for browser interoperability, DOM selection, and for various helper functions.

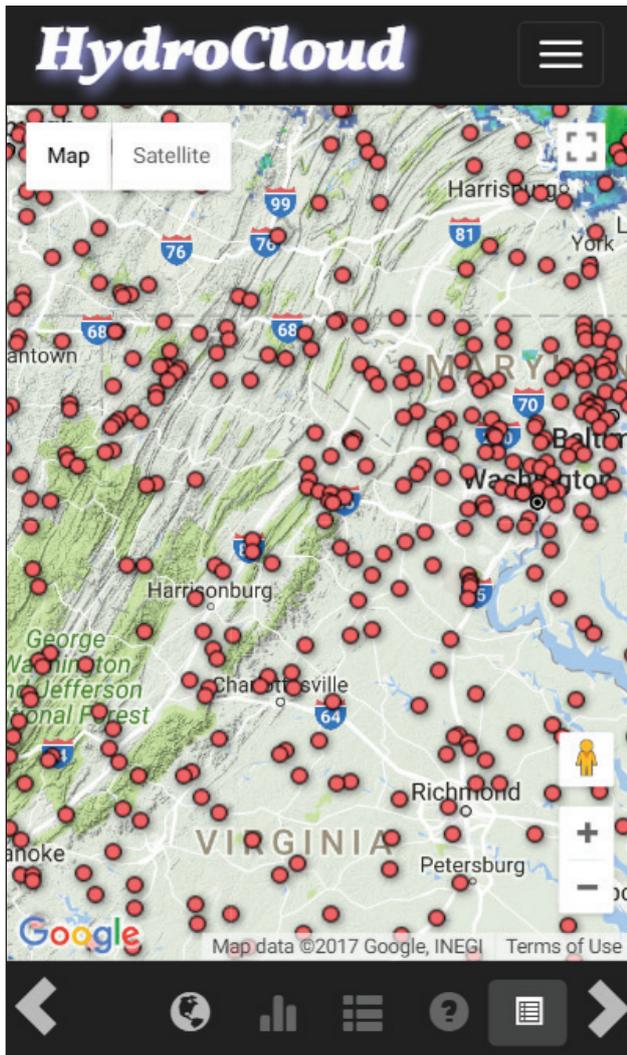
HydroCloud interacts with the user through three main views: the map view, the graph view, the stats

view, and the help view (Figure 1). The **map** view displays all of the spatial data used in the application, and is the primary interface for requesting additional data. The **graph** view displays temporal stream discharge data from the various providers. The **stats** view displays non-spatial data describing watershed characteristics. Supplementing the three data views are a static **help** view, which provides descriptions of how to use the application and links to the GitHub repository; a legend, which provides a floating window listing the selected sites; and a Chat sidebar where users can send messages to other users and contact the developers with questions.

The map view for HydroCloud (Figure 2) allows users to work with several layers of data, including a map or satellite imagery background, NEXRAD precipitation imagery, and points for each of the stream gage sites. All three of these layers are displayed using the Google Maps API, and are displayed using map tiles served by Google (base map), Iowa State University's Environmental Mesonet (NEXRAD imagery <https://mesonet.agron.iastate.edu/ogc/>), and Google Fusion Tables (the site point data). Users are able to use screen gestures to zoom and pan, and the standard Google Maps icon interface is also available for users to switch base maps, access Google StreetView, and for zoom functionality.



**Figure 1:** Diagram of the HydroCloud System. Users access spatial, non-spatial, and time-series data from a variety of external providers by interacting with the View layer. The ViewModel stores the data locally and maintains the state of the interface.



**Figure 2:** HydroCloud map view on a small mobile screen (320 × 540). Icons at the bottom of the screen indicate which view is currently active and allow users to change between the four major views and turn on the floating legend.

One important feature of the map view is the site location layer, which maps the location of thousands of stream gages as clickable point symbols. Clicking on a point opens an info window with the site name and some information about the site, such as (for US sites), the size of the watershed and the percent impervious landcover. Clicking on a point also triggers a request for stream discharge data and additional site data used in the graph and stats views. Although the USGS serves data for more than 1.6 million locations, we limited the map display in the US to the 9,322 sites included in the Gages-II dataset [11, 12], which we use as a source of non-spatial data. Similarly, we limited our display from other data providers to active sites, including 641 sites from the German WSV. For the most current list of sites and data sources, please visit <http://mroberge.github.io/HydroCloud/station-list.html>.

Unfortunately, even with ‘only’ 9,000 point locations, the resulting file sizes and network traffic would slow

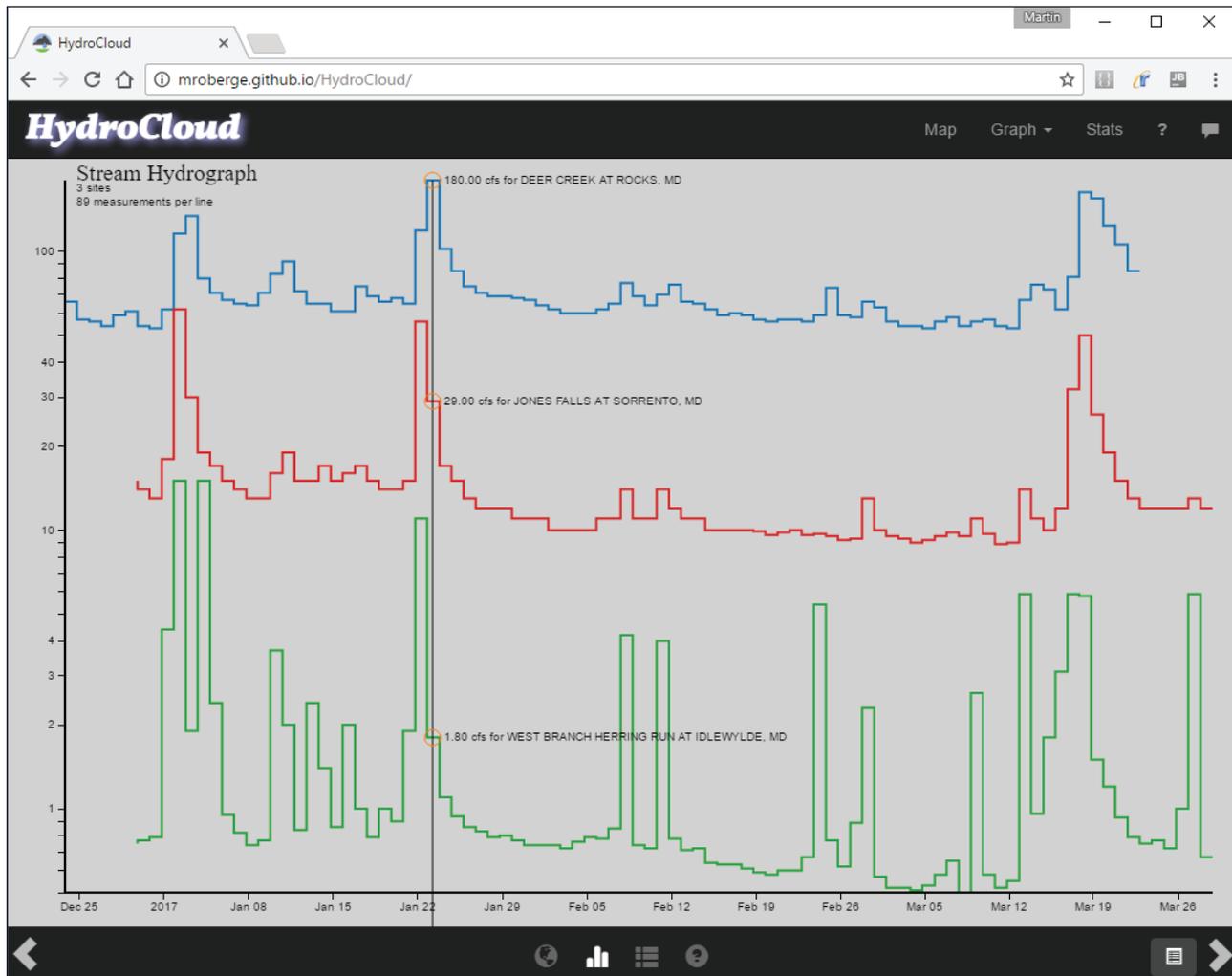
the application down considerably if the points were handled as a single JSON, XML, or KML file. Instead, we used the Google Fusion Table service to store a reduced version of the Gages-II dataset that only included a small subset of the hundreds of fields available. The Fusion Table service responds to non-spatial queries, and automatically creates and serves a set of mapping tiles for spatial queries. These tiles are largely transparent so that they can overlay the background and NEXRAD tiles, but they show the point locations of the stream gages. Like the other mapping tiles, they reduce network traffic and memory by only loading when needed in response to zoom or pan actions. Google Fusion Tables is currently free, with no limit to the number of queries it will serve. However, it is possible to replace this service either with a JSON file as described above, or by creating map tiles of the points and delivering them with a map tile server.

The graph view (**Figure 3**) was produced using Data-Driven Documents (D3js.org). This JavaScript library allows the creation of interactive, responsive graphs from large datasets with thousands of points. After a user clicks on a site in the map view, a request is submitted to the USGS Water Web Services for the past 90 days of daily mean stream discharge for that site. Once this data is returned, it is displayed as an interactive stream hydrograph, with discharge plotted on a logarithmic Y axis, and time on the X axis. A tool tip displays the discharge for each of the selected streams for a given date, allowing comparisons. The discharge data are plotted with missing segments for points that have error flags or are missing. The data are also plotted in such a way as to preserve the meaning of the data: daily mean discharge data is plotted as a horizontal line for the 24-hour period that was included in the average, while instantaneous values are plotted as connected points. The user can also display the data as a Flow Duration chart, which alters the X axis into a cumulative probability distribution. Future improvements include an interface for users to alter the time period of the request.

HydroCloud uses the HTML 5 localStorage API to store stream gage data locally. Each time the user requests data through the map interface, HydroCloud first checks to see if the data has already been requested and saved locally. If the data is present, then it is taken out of storage and added to the ViewModel. If the data is not present, or if is considered invalid or out of date, then the request for data is sent to the USGS. The localStorage protocol works using key-value pairs, with HydroCloud using site id as the key, and the array of stream gage data as the value. This storage space is persistent between sessions, so that a user can access the data many days later without explicitly saving anything.

#### Quality control

Unit tests for HydroCloud are written with Jasmine, and are located in the main repository. They can be run by



**Figure 3:** Full-size graph view. Each line represents data from a different USGS stream gage. The tooltip shows the discharge and name for each gage.

visiting the test runner page, <http://hydrocloud.org/test>. This page also contains a link to another page that tests the availability of the various data services that HydroCloud draws upon. Bug reports and an issue-tracker are located at <https://github.com/mroberge/HydroCloud/issues>.

## (2) Availability

### Operating system

HydroCloud is compatible with modern, standards-compliant browsers, including Firefox, Chrome, Safari, Opera, IE9+, Android and iOS.

### Programming language

JavaScript, HTML5.

### Additional system requirements

None.

### Dependencies

Data Driven Documents (D3.js), JQuery, bootstrap, knockout, Google Maps API.

### List of contributors

Martin Roberge, Lead Programmer.  
Michael McGuire, System Designer.  
Jie Lian, Programmer.

### Software location

#### Archive

**Name:** Zenodo

**Persistent identifier:** DOI: 10.5281/zenodo.494978

**Licence:** MIT

**Publisher:** Martin Roberge

**Version published:** v0.3.0

**Date published:** 04/04/2017

#### Code repository

**Name:** GitHub

**Identifier:** <https://github.com/mroberge/HydroCloud>

**Licence:** MIT

**Date published:** 1/17/2014

### Language

English.

### (3) Reuse potential

The HydroCloud software is modular, documented, and meant to be easily reused. The main graphing function was written to the “Reusable Charts” standard set by the author of D3.js, Mike Bostock[13]. This standard provides a uniform API for calling and using a chart. Developers that follow this standard can incorporate their own charts into HydroCloud simply by adding a link to the main page to call their code. Instructions for developers can be found at: <https://github.com/mroberge/HydroCloud/wiki/Contribute-to-HydroCloud>.

Users who want to incorporate the hydrographs used by HydroCloud into their own webpages can do so without knowing anything about how the internal code operates. Functionality such as controlling the graph size, adding data, or using a logarithmic Y scale can all be controlled using getter and setter methods. Directions for using the code are located at: <https://github.com/mroberge/HydroCloud/wiki/Add-a-hydrograph-to-your-website> and are also located in comments in the code itself.

Users can also incorporate the entire HydroCloud website into their own web pages using `<iframe>`. This technique can be used in conjunction with Jupyter Notebooks. Instructions for this are located at: <https://github.com/mroberge/HydroCloud/wiki/Embed-HydroCloud>.

Support for HydroCloud is provided through an integrated Gitter chat page that allows users to instantly contact the developer and other users. This feature is activated by tapping a chat icon, which opens an active chat window. This feature is meant to make it possible for users to highlight interesting sites, discuss hydrology, suggest new features, and ask for help.

Developers who are interested in extending the functionality of HydroCloud can do so through the normal GitHub process, using the issues list to file bug reports, suggest new features, or submit new sources of data. The help page and README file contain a feature list, examples of how to adapt HydroCloud to other uses, and descriptions of how to contribute code by submitting a pull request.

#### Competing Interests

The authors have no competing interests to declare.

#### References

1. **US Geological Survey** 2017 *USGS Water Data for the Nation* (online). National Water Information System: Web Interface. Available at: <https://waterdata.usgs.gov/> (Accessed 12 June 2017).
2. **Wasserstraßen- und Schifffahrtsverwaltung des Bundes** 2017 *PEGELONLINE* (online). Gewässerkundliches Informationssystem. Available at: <http://www.pegelonline.wsv.de> (Accessed 12 June 2017).
3. **UK Environment Agency** 2017 *Real-time API reference* (online). Environment Agency Read Time flood-monitoring API. Available at: <https://environment.data.gov.uk/flood-monitoring/doc/reference> (Accessed 12 June 2017).
4. **Commonwealth of Australia Bureau of Meteorology** 2017 *Water Data Online: Water Information* (online). Water Data Online. Available at: <http://www.bom.gov.au/waterdata/> (Accessed 12 June 2017).
5. **Open Geospatial Consortium** 2015 *OGC WaterML* (online). [opengeospatial.org](http://www.opengeospatial.org) Available at: <http://www.opengeospatial.org/standards/waterml> (Accessed 28 March 2017).
6. **Valentine, D** and **Zaslavsky, I** 2009 *CUASHI WaterML 1.0 Specification* (online). CUASHI. Available at: [http://his.cuahsi.org/documents/WaterML\\_1\\_0\\_part1.pdf](http://his.cuahsi.org/documents/WaterML_1_0_part1.pdf) (Accessed 28 March 2017).
7. **Ames, D P, Horsburgh, J S, Cao, Y, Kadlec, J, Whiteaker, T** and **Valentine, D** 2012 HydroDesktop: Web Services-Based Software for Hydrologic Data Discovery, Download, Visualization, and Analysis. *Environmental Modelling & Software*, 37: 146–156. DOI: <https://doi.org/10.1016/j.envsoft.2012.03.013>
8. **McGuire, M P, Roberge, M C** and **Lian, J** 2015 Channeling the Water Data Deluge: A System for Flexible Integration and Analysis of Hydrologic Data. *International Journal of Digital Earth*. DOI: <https://doi.org/10.1080/17538947.2015.1031715>
9. **McGuire, M P, Roberge, M C** and **Lian, J** 2014 HydroCloud: A Cloud-Based System for Hydrologic Data Integration and Analysis. *COM.Geo 2014 Proceedings on Computing for Geospatial Research and Application*. IEEE Xplore Digital Library. DOI: <https://doi.org/10.1109/COM.Geo.2014.16>
10. **McGuire, M P** and **Roberge, M C** 2014 The Design of a Collaborative Social Network for Watershed Science. In *Proceedings of the 2nd International Conference on Geo-Informatics in Resource Management & Sustainable Ecosystems* (GRMSE 2014). October 3–5, 2014, Ypsilanti, Michigan, USA. DOI: <https://doi.org/10.1007/978-3-662-45737-5>
11. **Falcone, J** 2011 *GAGES-II: Geospatial Attributes of Gages for Evaluating Streamflow* (online). US Geological Survey, Reston, VA. Available at: [https://water.usgs.gov/lookup/getspatial?gagesII\\_Sept2011](https://water.usgs.gov/lookup/getspatial?gagesII_Sept2011) (Accessed 28 March 2017).
12. **Falcone, J, Carlisle, D M, Wolock, D M** and **Meador, M R** 2010 GAGES: A stream gage database for evaluating natural and altered flow conditions in the conterminous United States. *Ecological Archives*, 91(2): 621. DOI: <https://doi.org/10.1890/09-0889.1>
13. **Bostock, M** 2012 *Towards Reusable Charts* (online). Available at: <https://bost.ocks.org/mike/chart/> (Accessed 28 March 2017).

**How to cite this article:** Roberge, M C, McGuire, M P and Lian, J 2017 HydroCloud: A Web Application for Exploring Stream Gage Data. *Journal of Open Research Software* 5: 18, DOI: <https://doi.org/10.5334/jors.173>

**Submitted:** 04 April 2017    **Accepted:** 16 June 2017    **Published:** 23 August 2017

**Copyright:** © 2017 The Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 International License (CC-BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited. See <http://creativecommons.org/licenses/by/4.0/>.