

Create Awesomeness: Use Custom Visualizations to Extend SAS® Visual Analytics to Get the Results You Need

Robby Powell and Renato Luppi, SAS Institute Inc., Cary, NC

ABSTRACT

SAS® Visual Analytics is an excellent visualization and analytics tool, but sometimes you need to add custom functionality to create the exact tool, content, or analysis your team needs. This paper looks at how to extend SAS Visual Analytics 8.2 by adding your own custom visualizations. It shows the integration of JavaScript visualization libraries such as D3 through the SAS Visual Analytics data-driven content object. You will be able to extend SAS Visual Analytics with your own custom functionality to create the customized visual analytics experience your team needs for business intelligence and data-driven decisions.

INTRODUCTION

Although SAS Visual Analytics includes a wide variety of objects for visualization, analytics, and reporting, sometimes you might want to integrate external content to customize your reports with your own functionality. This functionality might be for graphing or for other capabilities to meet the exact needs of your audience. You can extend SAS Visual Analytics and SAS® Viya® in various ways: adding functionality inside SAS Visual Analytics, embedding SAS Visual Analytics content inside another site or application, or using APIs to communicate with SAS® Cloud Analytic Services (CAS). The focus here is on adding customized visualization capabilities to your SAS Visual Analytics content through the data-driven content object.

This paper looks at popular JavaScript visualization libraries, discusses how to embed these libraries with the data-driven content object, shows how SAS Visual Analytics data is passed between SAS Visual Analytics and JavaScript libraries, and provides an end-to-end implementation of a SAS Visual Analytics exploration using a D3 JavaScript library. This paper aims to get you started integrating SAS Visual Analytics with external visualization libraries.

DATA-DRIVEN CONTENT

So, what is it? Data-driven content (or DDC, for short) is a new report object in SAS Visual Analytics 8.2 that enables you to create your own custom visualization based on JavaScript and to embed it in SAS Visual Analytics reports. Like other SAS Visual Analytics objects such as bar charts, pie charts, and more designed for use with advanced analytics, you can interact with DDC objects as the source or target of actions. You can assign roles and apply filters and ranks to DDC objects, making your custom visualization act as if it were a native SAS Visual Analytics visualization. In other words, you can use DDC to expand the already extensive number of charts and plots that SAS Visual Analytics offers by leveraging commercial or open-source JavaScript frameworks for visualization, such as D3.js, C3.js, Google Charts, Chart.js, dimple.js, dygraphs, jqPlot, Flot, and so on.

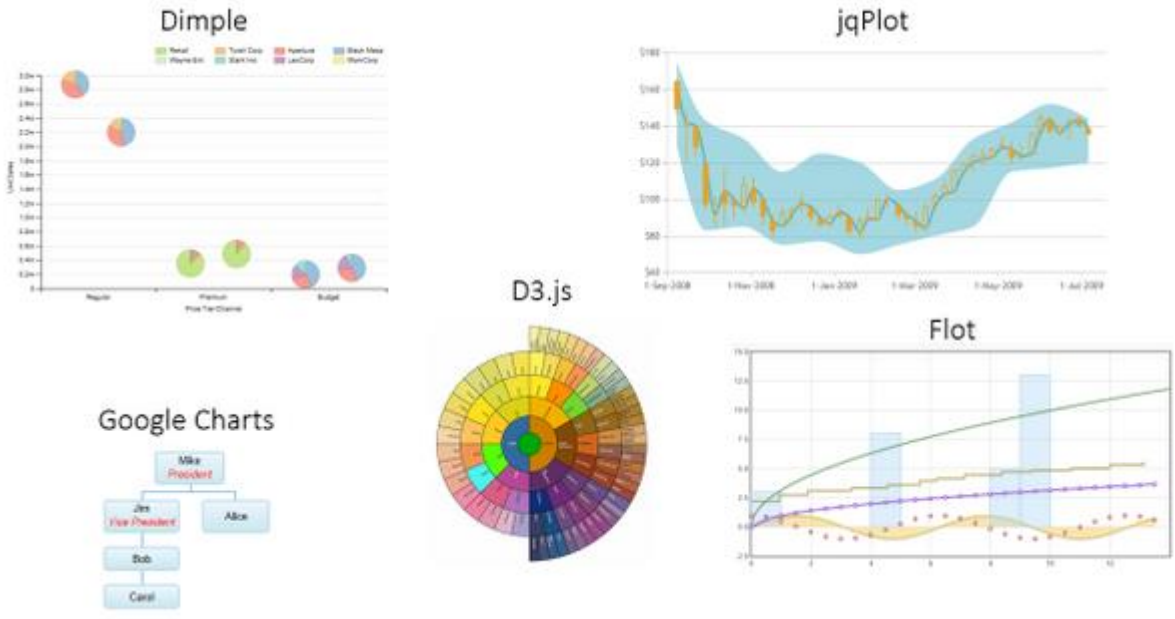


Figure 1. Example of third-party JavaScript visualization frameworks

Hopefully, this doesn't come as a surprise, but complete integration between SAS Visual Analytics and JavaScript libraries requires JavaScript skills. The goal of this paper is not to teach you JavaScript, but to highlight the key components that you, the third-party visualization developer, must know to be able to seamlessly integrate your visualization with SAS Visual Analytics.

INTRODUCTION TO SAS VISUAL ANALYTICS DATA-DRIVEN CONTENT OBJECT

To get started, you need to drag and drop the data-driven content object onto a SAS Visual Analytics report and assign a few data columns to it. The default visualization that you see is a table that is rendered from the following URL:

http://<your.viya.host>/SASVisualAnalytics/resources/custom_table.html

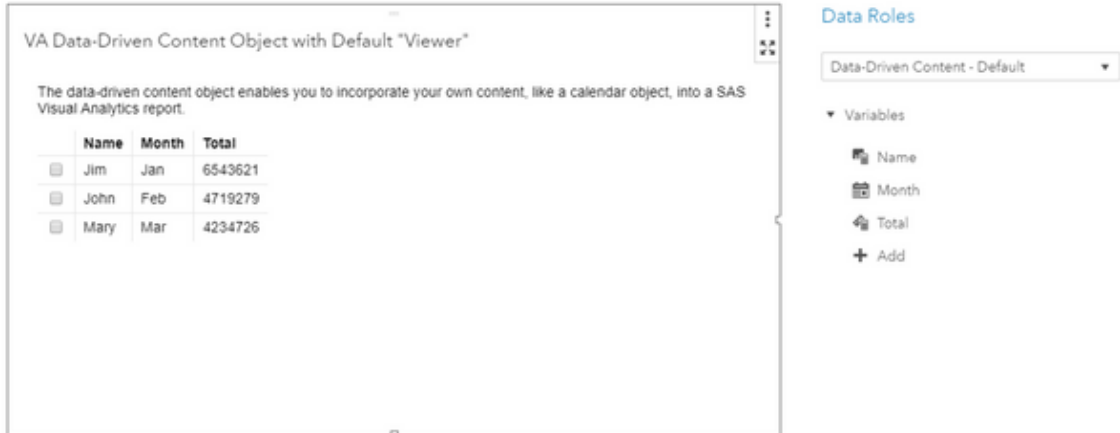


Figure 2. Default viewer for a data-driven content object

Don't overlook this default viewer because it illustrates the core concepts and capabilities supported by the data-driven content object. Even though this is a simple sample, the source data could have come from a much larger table. SAS Visual Analytics queries and aggregates the data using its in-memory engine and passes the results to the data-driven content object. So, even if you are using a third-party visualization, you are still leveraging SAS Visual Analytics computing power behind the scenes.

SETTING UP YOUR DATA-DRIVEN CONTENT OBJECT

The data-driven content object exists in the **Other** section in the **Objects** pane in SAS Visual Analytics. Drag and drop it onto your report as you would any object.

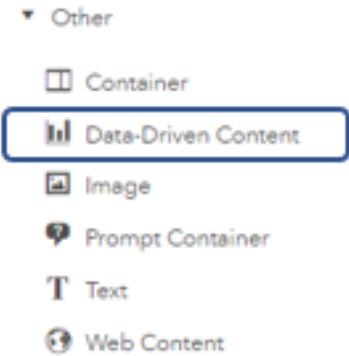
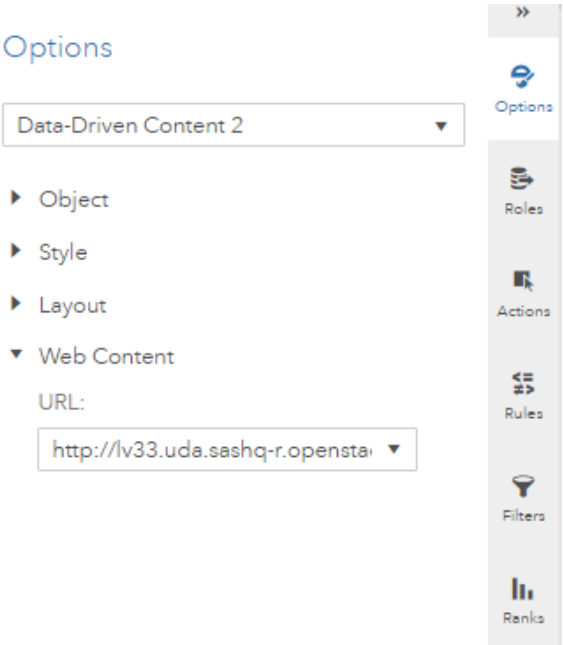


Figure 3. Data-driven content object in SAS Visual Analytics Objects pane

You can directly add the URL of the location where your data-driven content resides.



Or, you can add multiple custom, third-party, data-driven graphics by selecting **Edit administration settings** from the menu.

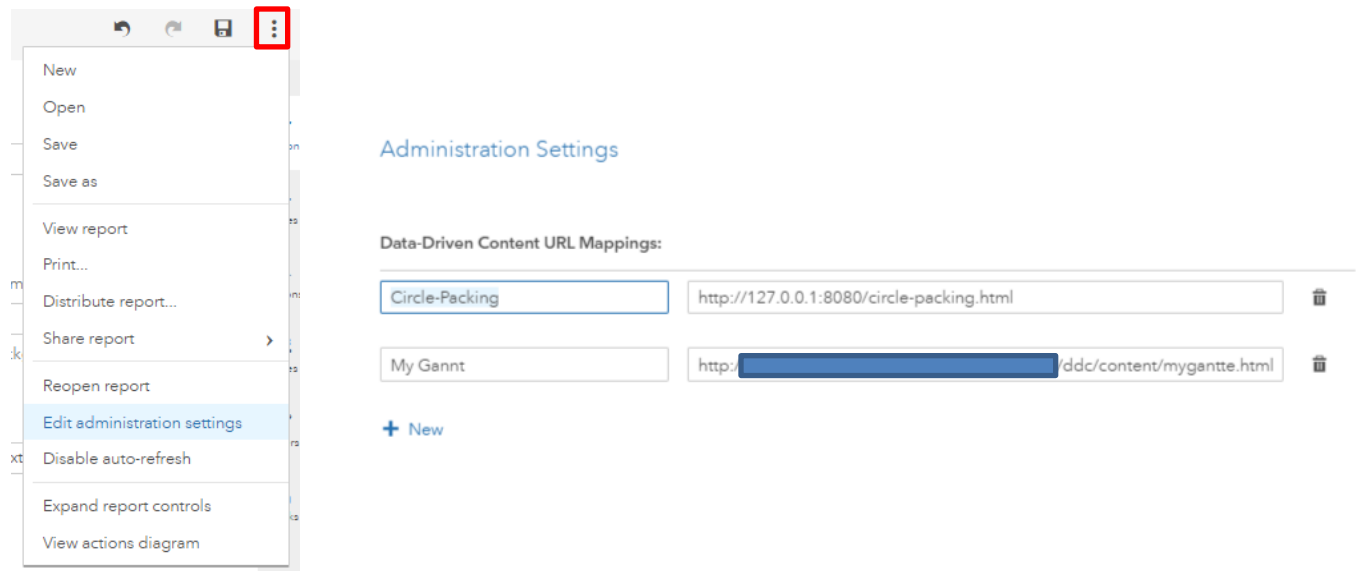


Figure 4. Data-driven content URL mapping

COMMUNICATING WITH THE DATA-DRIVEN CONTENT OBJECT

SAS Visual Analytics communicates with the data-driven content object through messages that include a JSON object containing the aggregated data items and a few other items. Each time there is a change to the data, such as applying a filter, SAS Visual Analytics sends a new message to the data-driven content object with a new JSON object.

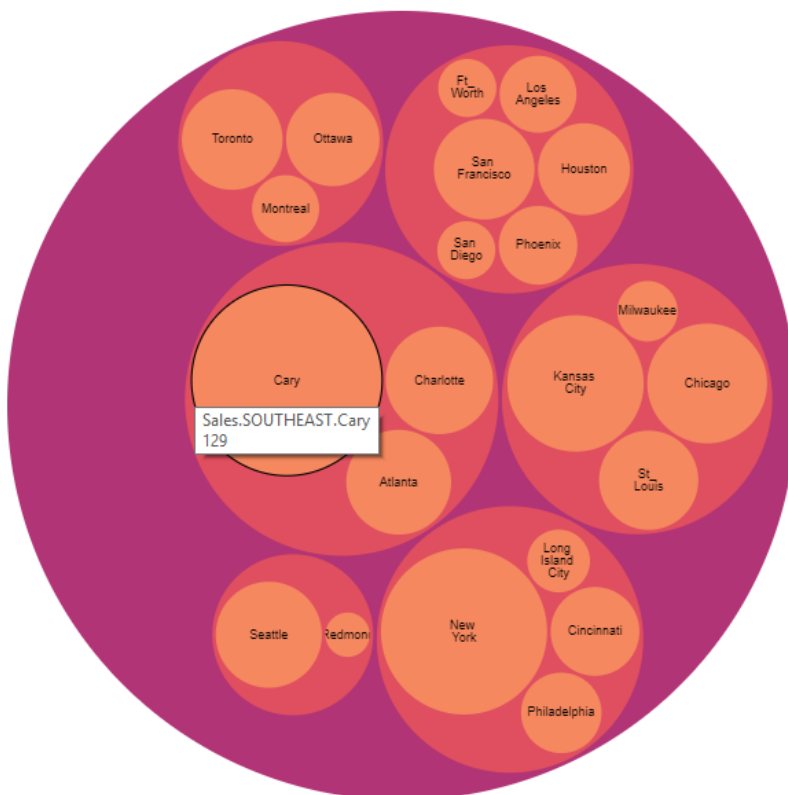


Figure 5. D3 circle-packing hierarchy graphic using data from SAS Visual Analytics with Cary showing the data tip

The data used to generate this graphic is from the circle-packing graphic. It is a JSON object that needs to be parsed into a data format that the D3 graphic can use.

```

0:(3) ["CANADA", "Montreal", 16]
1:(3) ["CANADA", "Ottawa", 31]
2:(3) ["CANADA", "Toronto", 36]
3:(3) ["CENTRAL", "Chicago", 51]
4:(3) ["CENTRAL", "Kansas City", 66]
5:(3) ["CENTRAL", "Milwaukee", 13]
6:(3) ["CENTRAL", "St_Louis", 35]
7:(3) ["NORTHEAST", "Cincinnati", 28]
8:(3) ["NORTHEAST", "Long_Island_City", 14]
9:(3) ["NORTHEAST", "New York", 98]
10:(3) ["NORTHEAST", "Philadelphia", 23]
11:(3) ["NORTHWEST", "Redmond", 7]
12:(3) ["NORTHWEST", "Seattle", 40]
13:(3) ["SOUTHEAST", "Atlanta", 39]
14:(3) ["SOUTHEAST", "Cary", 129]
15:(3) ["SOUTHEAST", "Charlotte", 41]
16:(3) ["SOUTHWEST", "Ft_Worth", 12]
17:(3) ["SOUTHWEST", "Houston", 30]
18:(3) ["SOUTHWEST", "Los Angeles", 21]
19:(3) ["SOUTHWEST", "Phoenix", 22]
20:(3) ["SOUTHWEST", "San Diego", 12]
21:(3) ["SOUTHWEST", "San Francisco", 36]

```

Figure 6. Data received from SAS Visual Analytics as a JSON object and used to create the graphic in Figure 5

Additional information in the JSON object includes brushing selection. Filtered data is passed as well, but it is through the standard data feed JSON object, as shown in Figure 7.

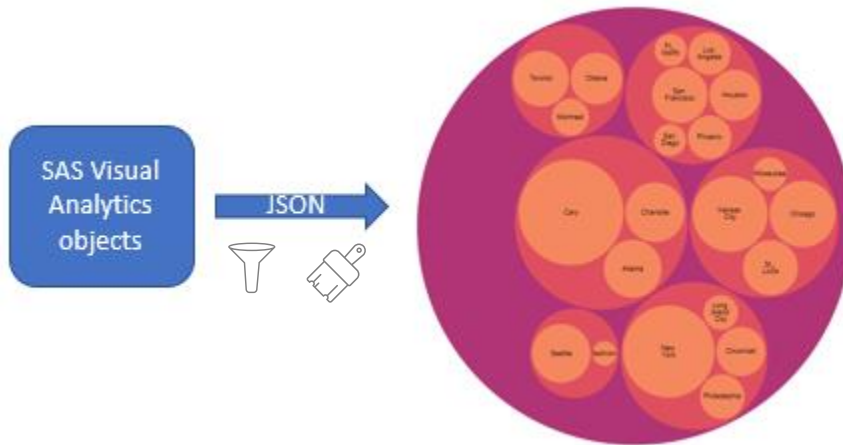


Figure 7. Initial data goes from SAS Visual Analytics objects to D3 objects as JSON objects

Data brushing is different because it selects data elements. This information is sent through the same JSON object with a column indicating selection state. Notice that **Calgary** is the selected city in the following data as denoted by the non-zero value in the fourth position in the array that includes "Calgary":

```

0:(4) ["CANADA", "Alliston", 31, 0]
1:(4) ["CANADA", "Burlington", 18, 0]

```

```

2:(4) ["CANADA", "Calgary", 17, 1]
3:(4) ["CANADA", "Don Mills", 15, 0]
4:(4) ["CANADA", "Kirkland", 13, 0]
5:(4) ["CANADA", "Montreal", 16, 0]
6:(4) ["CANADA", "Ottawa", 31, 0]
7:(4) ["CANADA", "Rexdale", 15, 0]
8:(4) ["CANADA", "Toronto", 36, 0]

```

Figure 8. Data received from SAS Visual Analytics specifying “Calgary” as the selected item by the “1” in the fourth position in the array. All other fourth-position values are “0” indicating “not selected”.

One of the coolest features of SAS Visual Analytics is by-directional selection and filtering of data across multiple visualization objects. The data-driven content object supports this feature by sending messages back to SAS Visual Analytics. Selection messages and instructional messages can both be sent. To send selection messages, you must pass a JSON structure from the JavaScript library back to SAS Visual Analytics. You add code to your JavaScript library to handle events that your third-party visualization generates, such as the user selecting specific data points. Instructional messages are text messages used to notify the user of something occurring within the custom visualization.

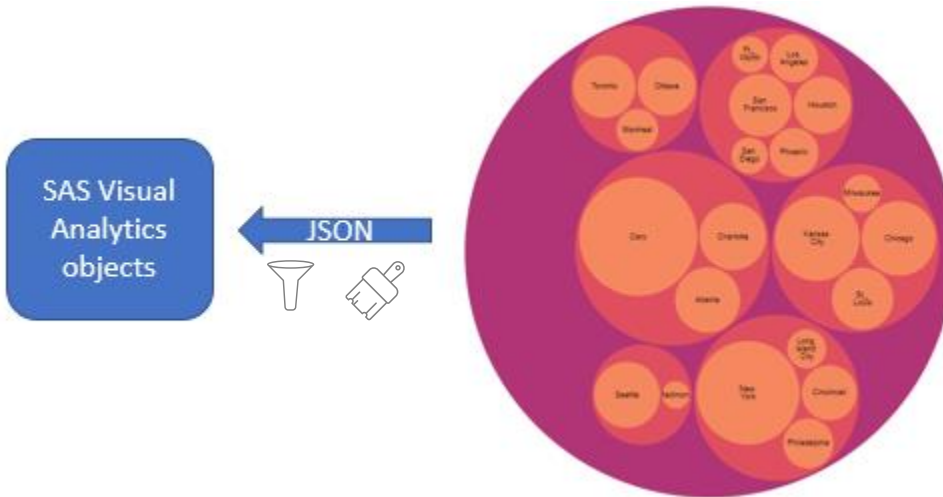


Figure 9. Data and information can be sent from a third-party visualization back to SAS Visual Analytics

It is important to remember that there are three different types of messages between SAS Visual Analytics and the data-driven content object, and they are all in JSON format.

From SAS Visual Analytics to third-party visualization (received from SAS Visual Analytics)

1. A data message that contains mainly data and columns metadata as well as other information

From third-party visualization to SAS Visual Analytics (sent to SAS Visual Analytics)

2. A selection message that is used to filter or brush other SAS Visual Analytics objects
3. An instructional message that is text to be displayed in the data-driven content object

CONCLUSION

You normally think about visualizations when dealing with data-driven content objects, but it's not restricted to visualizations only. There are implementations that use data-driven content objects to pass data and communicate with other applications. People will continue to come up with other interesting ideas. The sky is the limit!

To learn more about the data-driven content object, check out Renato Luppi's blog series, [Data-Driven Content](#). Full examples of third-party visualizations can be found in [GitHub](#). There isn't much in the library yet, but the idea is to get the library started so that everyone can contribute and benefit from new visualizations and samples. If you contribute new visualizations, please let us know by leaving a note in the Comments section.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the authors:

Robby Powell
100 SAS Campus Drive
Cary, NC 27513
SAS Institute Inc.
robby.powell@sas.com
<http://www.sas.com>

Renato Luppi
100 SAS Campus Drive
Cary, NC 27513
SAS Institute Inc.
renato.luppi@sas.com
<http://www.sas.com>

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.