

Paper 5027-2020

Raising the Bar! Developing World-Class Reports with SAS® Visual Analytics®

Lucy Smith, Amadeus Software Ltd

ABSTRACT

SAS® Visual Analytics 8.x provides around 40 different visuals out of the box. This gives all users, both novice and advanced, a lot of potential ways to present their data. However, as the reporting requirements and dashboards develop, the default visuals might not always be enough.

This paper covers three exciting ways to enhance and extend the default visuals in SAS Visual Analytics. The first method starts with default visual and explores options in SAS Visual Analytics to customize the appearance. The second method looks at using the custom graph builder to extend the default visuals to represent your data in the required way. The last method showcases how to include a completely new custom visual to SAS Visual Analytics, using the D3.js JavaScript framework for visualization.

This paper is for SAS Visual Analytics users of any ability, and the techniques covered are applicable in SAS Visual Analytics 8.x on SAS® Viya®.

INTRODUCTION

Whether you are a seasoned SAS Visual Analytics user or have just started using the tool, you will be familiar with the extensive list of out of the box visualizations that are available. The objects tab contains a comprehensive list of graphs to help you present your data in a suitable way. There are also additional objects, called controls, to help you create dynamic graphs and containers to control the layout of your report.

Data comes in many different formats so the perfect way to present it may not be possible using the default visuals. Fortunately, SAS Visual Analytics provides users with multiple techniques that can be used to customize the way your graphs look, even using charts that you have created outside SAS Visual Analytics.

Having these new techniques at your fingertips will ensure you can present your data in the most appropriate and effective way possible.

Each of the examples presented in this paper will use data from a case study focusing on a UK-based Gym company. SAS Visual Analytics is being used to report important business metrics for each of their gyms, including overall performance (total members at each gym), performance over time and performance compared to targets.

ENHANCE A DEFAULT VISUAL

The first technique that will be covered is using a visual that is already available in SAS Visual Analytics and exploring options that can be used to enhance it.

To create a perfect visual within a dashboard or report, it is important to explore all the additional options available with the graph you are using. By default, there are plenty of different graphs to use and the first task is to select the most suitable one for the type of data you have. For example, if you are trying to visualize categorical data, a bar chart may be suitable and if you have time-based data a line chart will be more appropriate.

Two charts (Figure 1) have been created to try to convey information about how many members each gym currently has. End users will be likely to ask the following questions:

- How many members does each gym have?
- Which gym has the most members?
- What is the difference in the number of members between each gym?

In Figure 1, on the left is an example of a pie chart, that in this case is an unsuitable chart to answer these questions. It does not provide an easy comparison between gyms and it **doesn't** contain a scale to show the number of members. It would very hard to answer the three questions from above using this graph alone.

The chart on the right gives a better visualization of the data. It is easy to see how many members each of the gyms have and it is easy to make comparisons between them.

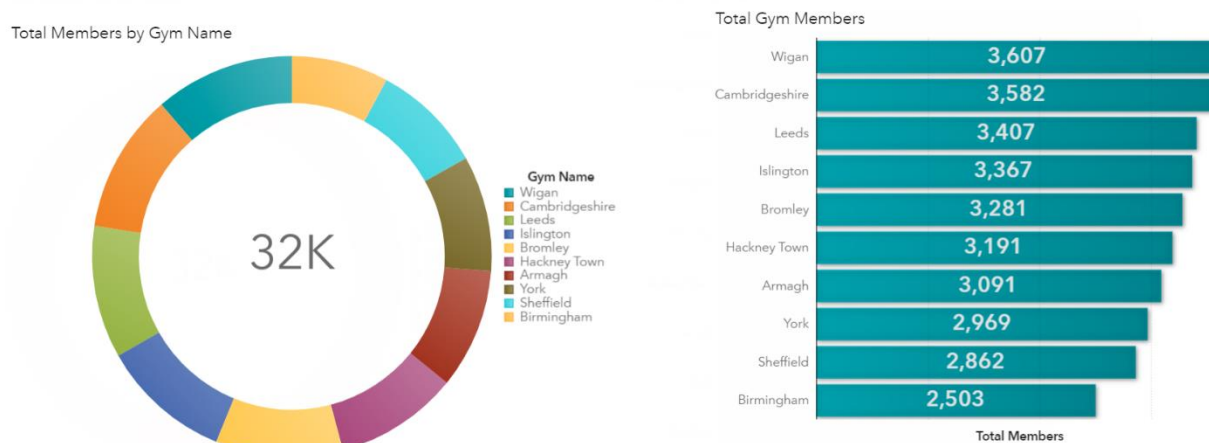


Figure 1. SAS Visual Analytics Example Charts

Once the type of graph has been selected, there are many other aspects related to the appearance of the graph that can be modified to visually enhance how information is presented to a viewer.

The main key point to remember is not to over complicate it. Ask yourself if you can obtain the key information you are trying to transmit in the first 5 seconds of seeing the image.

The following options and settings can help you customize default graphs. These are all accessible from the tabs displayed on the right-hand side of the page of SAS Visual Analytics:

- **Filters.** Filters should be used to ensure the information displayed is exactly what the viewer needs to know, removing any irrelevant information.
- **Ranks.** Use Ranks to limit the number of data points shown on the graph. Too much data on one chart can hide important information. The ranks tab can be used to show the top or bottom X (by count or percent). To ensure information is not lost to the viewer, an Info Window can be used with the graph to provide the full details.
- **Styles.** Select an appropriate color pallet for your graph. Default colors are selected to ensure no single data points stand out more than others. If you require specific data to stand out, exaggerate the color scheme. A report theme can also be used to ensure the whole report utilizes the same style.
- **Display Rules.** Display Rules can dynamically change the colors used in the graph by applying conditional rules. This will ensure that if the data values meet the given criteria, they are highlighted in the graph.

Targeted Bar Chart Example

The next graph (Figure 2) is trying to show which gyms are meeting their membership targets. This has been presented using the default targeted bar chart.

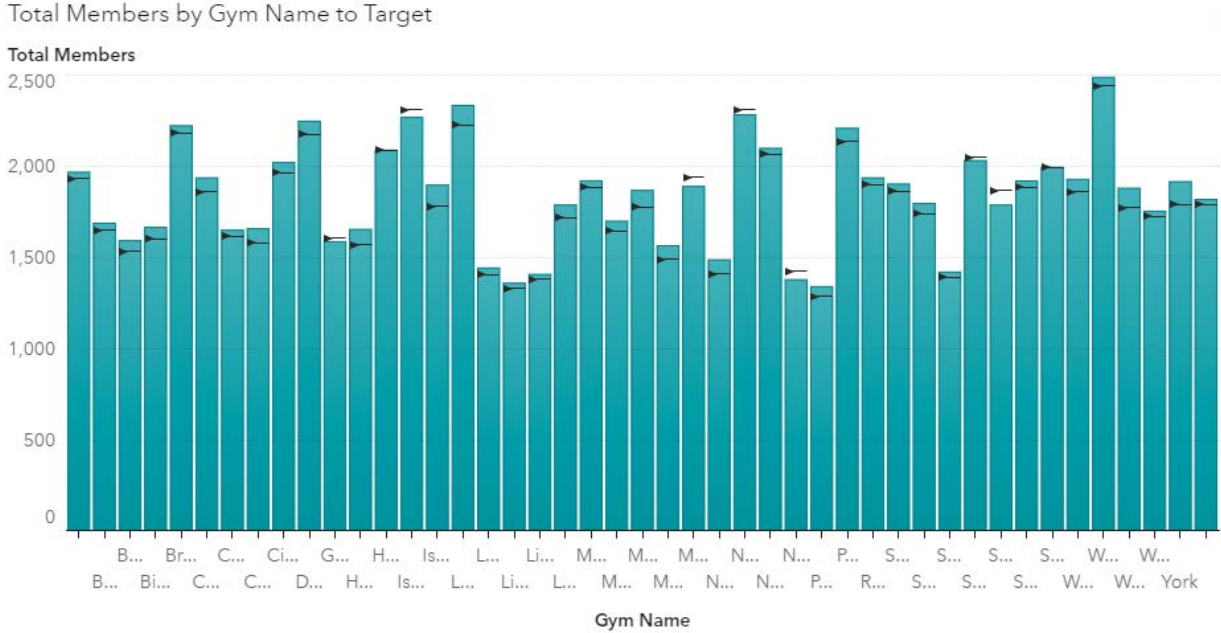


Figure 2. SAS Visual Analytics Default Targeted Bar Chart

However, from this graph we cannot quickly find which gyms are below their targets, clearly see the gym names or identify the worst-performing gyms.

This graph can easily be improved by using some of the options mentioned earlier. They are (Figure 3):

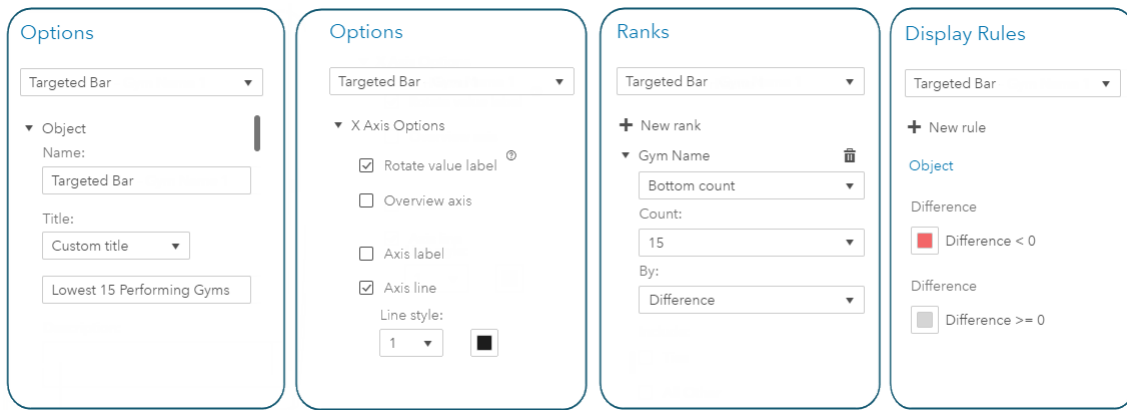


Figure 3. Options Available in the SAS Visual Analytics Targeted Bar Chart

- Options. Override default title to provide a more meaningful description of the information displayed. Modify the angle of the X-axis values so gym names can be read.
- Ranks. Limit the number of gyms to be displayed to 15, and only show the gyms that are furthest from meeting their membership targets.
- Display Rules. Ensure gyms that have not met their targets are set to a different color. This could also be the opportunity to incorporate company branding using styles.

The following graph (Figure 4) incorporates all these modifications. Red has been used to highlight the gyms that have not met their target for total members, ensuring they stand out.

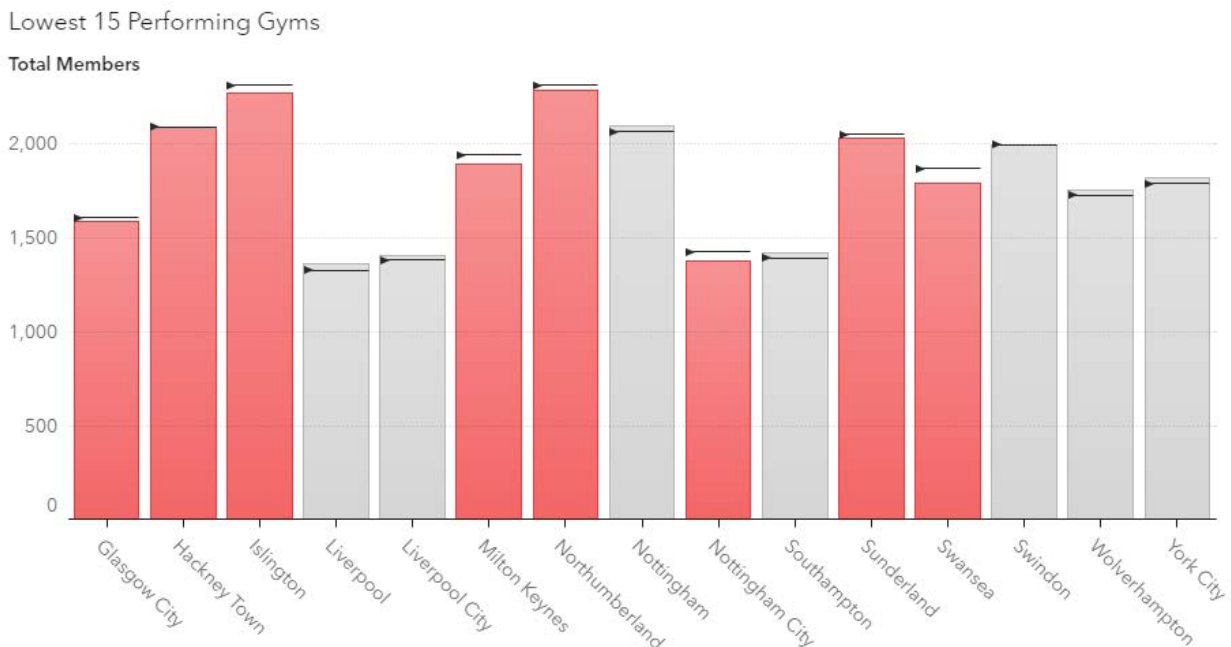


Figure 4. Improved SAS Visual Analytics Targeted Bar Chart

The steps described in this section are the first considerations to make when designing a new report. Customizing a graph using the options available is a simple way to create engaging and informative graphs.

SAS GRAPH BUILDER

The second technique covered will be using the SAS Graph Builder. There will be occasions when the default SAS Visual Analytics graphs can't display the data in the way you need them to, and in these cases, the task is often resolved with the SAS Graph Builder. The graph builder allows the user to overlay different graphs on the same or different axis and customize options that aren't seen in the default SAS Visual Analytics objects.

The new graph templates that you create can be saved and shared between all users who develop SAS Visual Analytics reports. Creating the graphs is a one-time task, after which they are used in the same way as the out of the box ones.

Custom Graph - Annotating a Line Chart

The gym company must generate reports each month to allow regional managers to quickly compare how well the gyms are performing over time. Comparisons to the same month in the previous year will be the focus of one of the reports, and a chart that helps fulfil this requirement is a line chart with text annotations on specific data points to highlight important months.

Two of the charts that are available to display this type of information are a Numerical Series Plot and a Line Chart. They both provide a similar output except for a few key differences. The Numerical Series plot expects a measure for the X-axis and you can add labels to the data points. The Line Chart expects a category or date for the X-axis and you cannot add labels to the data points.

As the graph that I am trying to design needs to present a date variable with labels on data points, the SAS Graph Builder can be used to combine the functionality of both charts.

Graph Elements

When you create a custom graph from the SAS Graph Builder application, the canvas in the middle of the screen is where the graph will be designed. The next steps will take you through the actions that are needed to create the custom graph:

1. Drag the Series Plot into the center. This will be used to present markers and labels for important time points (added in step 3).
2. Drag the Line Chart on top of the Series Plot. This will be used to display the monthly data.
3. A new role of *Data Label* needs to be added to the Series Plot (Figure 5), using Role Definitions. This will enable users to add a categorical label to any important data points in the line chart.

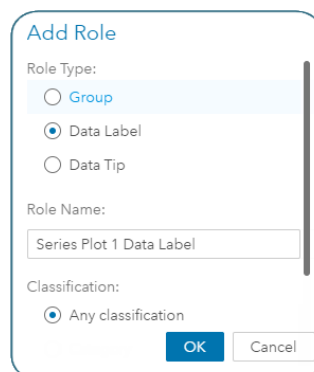


Figure 5. Define a New Role

- If this custom graph was used in a report, as it stands, the lines from both charts will be visible, the X-axis values will not be clear, and the legend will display both the line chart and the series plot which is duplicating information.

This is not a problem though, as all graphs designed in the SAS Graph Builder have a lot of customizable options to control how the graph looks when used in a report (Figure 6). For this example, it is important to only show the line from the Line Chart and the markers and label from the Series Plot. The legend will show repeating information so we will remove this as well. The following options can be used to implement these:

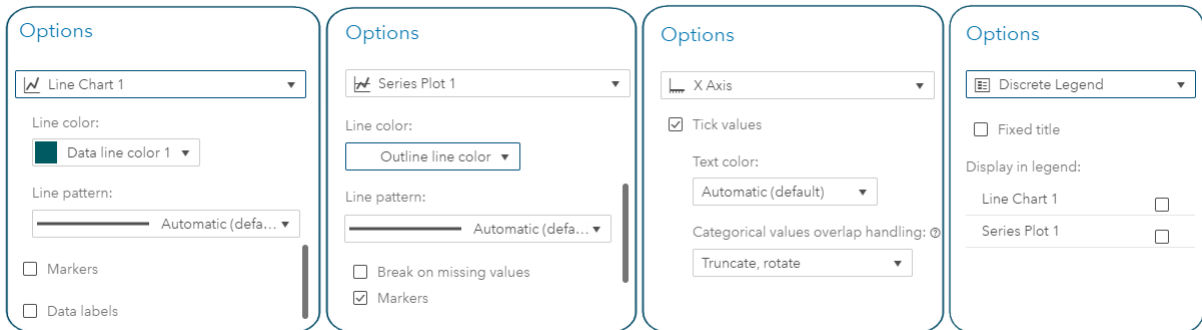


Figure 6. Options in SAS Graph Builder

- On the Line Chart 1 options, set the line color to *Data line color 1*.
- On the Series Plot 1 options, set the color to *Outline line color* so a second line is not displayed and check the Markers box to display time point markers.
- On the X-axis options, under Tick values and Categorical values overlap handling, select *Truncate, rotate*. This will rotate the values displayed so that they can all be read.
- On the Discrete Legend, uncheck both charts to remove the legend.

Once the graph has been created, save the new template in a location that all required users can access. Custom graphs can be added to the available objects tab by importing them. After the import, the new template will appear into your objects pane, allowing it to be used in any future report as if it were a native SAS Visual Analytics chart (Figure 7).

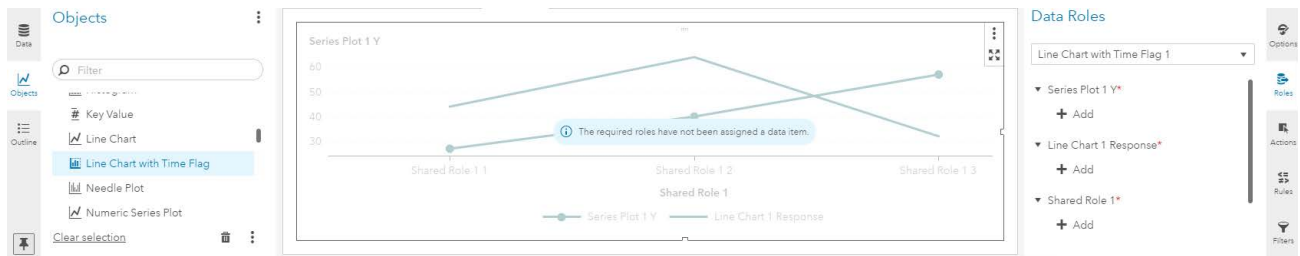


Figure 7. Custom Chart in SAS Visual Analytics

As the graph created in the SAS Graph Builder can be treated as if it were a native SAS Visual Analytics graph, the last step required is to define the data needed to populate the roles.

To present the total number of members the gym company has each month, the following data roles need to be assigned (Figure 8):

- Series Plot 1 Y. This role will have a measure that contains the total members, but only for the months that will be annotated (in this example, June).
- Line Chart 1 Response. This role will have the total number of members.
- Shared Role 1. This will be the X-axis date variable and is shared by both the Series Plot and the Line Chart.
- Series Plot 1 Data Label. This will be a new calculated column. It will contain the date, but only if the month is June. This will be used to flag those months on the graph.

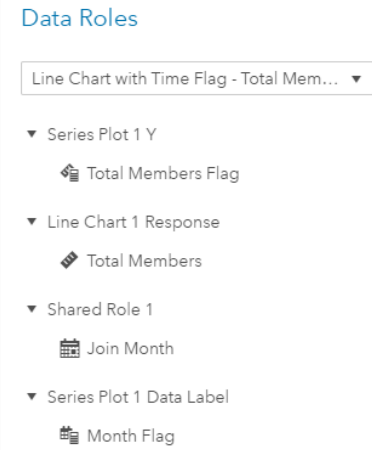


Figure 8. Custom Chart Roles

Once the roles have been assigned, the chart will display a line chart with labels for the key month of June (Figure 9):

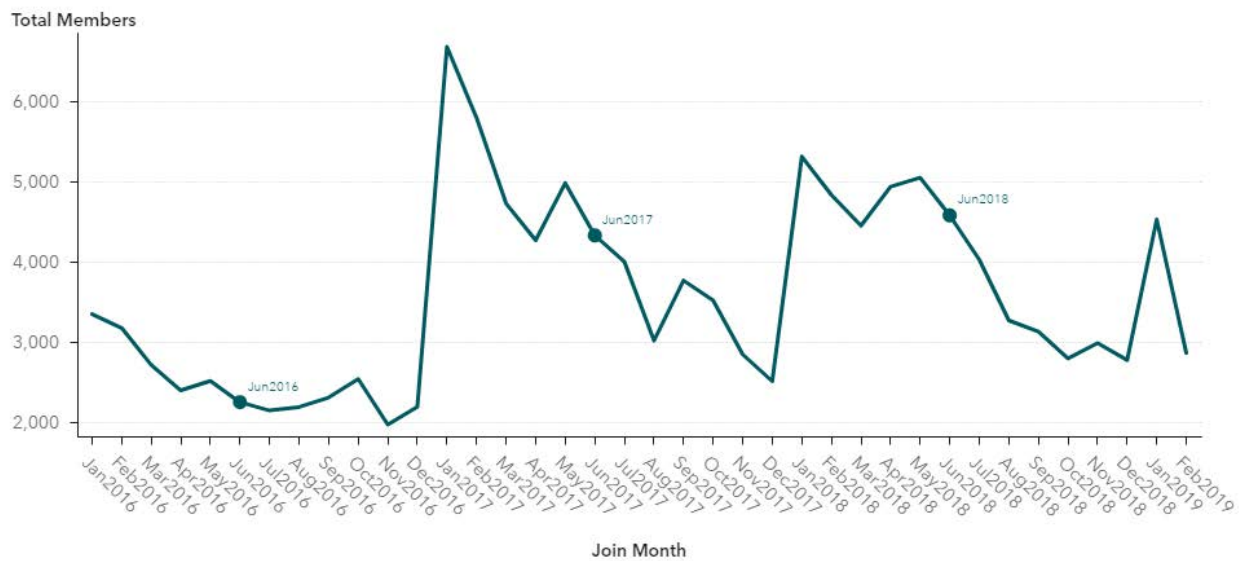


Figure 9. Custom Line Chart with Annotations

The SAS Graph Builder can create graph templates that overcome some limitations of the default graphs in SAS Visual Analytics. They allow the user to merge functionality of existing SAS Visual Analytics graphs to define the exact template they require.

DATA-DRIVEN CONTENT, D3

The third technique to extend the capability of SAS Visual Analytics is creating a graph outside of SAS Visual Analytics and incorporating it inside your report. This technique should be considered if there is not a suitable default graph or one that can be created using the SAS Graph Builder, as it requires some knowledge of JavaScript.

The Data-Driven Content object enables you to embed a JavaScript based visual inside a SAS Visual Analytics report. The JavaScript based visual needs to be in an HTML file, and can use one of many JavaScript libraries, including GoogleCharts, C3.js and D3.js.

The external visualizations will blend into your report and look like a native SAS Visual Analytics object. For a report creator, the object will be used in the same way as any other object, including using the Roles tab to define the data to be displayed. Much more of the functionality you would expect with a default graph will also be available, including filtering, interactions with other objects and creating links directly to other parts of your reports. This paper will focus on the D3.js library.

D3.js

D3.js is one of the many open source JavaScript libraries that are available to use. To create a new graph from scratch you will need to know some JavaScript, however D3 is not too complicated to learn if you have prior knowledge of HTML. There are also plenty of graphs that have already been created and shared. These can be found at <https://d3js.org/> and <https://github.com/d3/d3/wiki/Gallery>.

Interesting uses of D3.js include both creating completely new graphics or changing the functionality on types of graphs that are available in SAS Visual Analytics.

In SAS Visual Analytics, there are many ways to interact with a graph. In a Bar chart for example, the user can hover over a bar to see a tool tip, click a bar so see its interaction with another plot and double click to link to another page in the report.

In D3.js, there are ample opportunities to customize the actions that occur when a user hovers or clicks on a graph. Imagine that we are trying to create a graph that changes color when **the user's mouse hovers** over a part of the chart. This will not be possible using the objects present in SAS Visual Analytics but can be achieved by using a D3.js external graph.

The following example will discuss creating a graph that shows the total number of members at a range of gyms. The aim will be to create a graph such that when the user moves their mouse over a specific gym, the color of the surrounding gyms instantly update, based on their performance compared to the gym in question.

The following steps will cover the considerations needed when using D3 to ensure it can embed seamlessly and interact with your report in SAS Visual Analytics.

Create Your Graph

Designing your graph will require the most knowledge outside of SAS, but as previously mentioned, you can start with a graph that has already been created. The creation of a custom graph is outside the scope of this paper and we will use the d3.js example BarChartD3.html chart that can be found on GitHub <https://github.com/sascommunities/sas-global-forum-2020/tree/master/papers/5027-2020-Smith>.

Any D3.js graph can be embedded inside a SAS Visual Analytics report but, by default, the graph can only be rendered with external data. To allow the Data-Driven Content object to provide the data to the D3.js graph, a few programming considerations are to be made.

Listener

To fully embed a third-party visualization in SAS Visual Analytics, two considerations are needed. SAS Visual Analytics needs to pass the data to the third-party visualization and the third-party visualization needs to then use the data and render the graph.

SAS Visual Analytics shares data with third party visualizations using an HTML API [SAS INSTITUTE INC., 2019]. To receive the data from SAS Visual Analytics, the third-party visualization needs to be programmed to listen for it.

A listener is a piece of code that can be included in your HTML script and it will listen for data being passed from the Data-Driven Content object in SAS Visual Analytics.

After adding the code for the listener, you also need to utilize the data received in your D3.js code.

Once these steps have been implemented, any time the user interacts with the third-party visualization, the listener reacts immediately and renders the new graph.

Importing to SAS Visual Analytics

When your third-party visualization is ready for use, it will need to be saved on your SAS Viya server in the folder `/var/www/html`. When your HTML file is saved in this location, you will be able to reference it using your usual SAS Viya hostname URL followed by the name of the file. For example:

HTML file name: BarChartD3VA.html
SAS Viya full hostname: <https://viya.demo.com>.

As the HTML file is saved in the folder `/var/www/html` you can now reference it using the URL <https://viya.demo.com/BarChartD3VA.html>. This will be the URL you will need for your Data-Driven Content object.

Adding the chart in SAS Visual Analytics

To add the third-party visualization to a report in SAS Visual Analytics, you will need to follow the steps below in Figure 10:

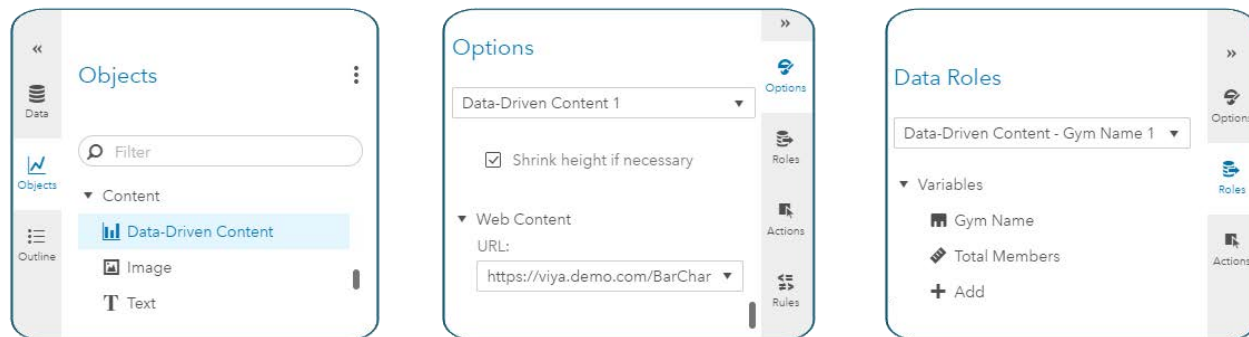


Figure 10. Custom Chart Roles

1. Add the Data-Driven Content object to the report.
2. In the objects tab, change the URL to include the name of your Viya server and object created, in this example <https://viya.demo.com/BarChartD3VA.html>.
3. In the Roles tab, add the required data (which will depend on the HTML and graph created). For this example, we are selecting the Gym name as a category and the Total number of Members as the measure.

The graph will then be rendered in your report. The graph will automatically be able to interact with other objects, in the example below (Figure 11) a button bar control has been added and is used to filter the graph by the type of membership. A rank has also been added to filter the graph to 25 gyms with the fewest members.

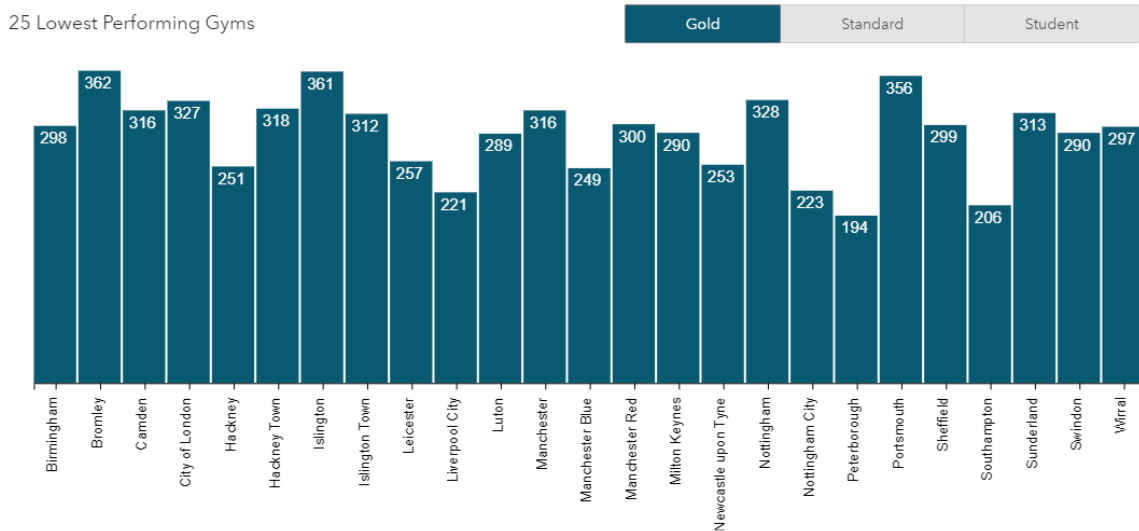


Figure 11. D3.js Bar Chart in SAS Visual Analytics

The new added graph is now dynamic, and as the mouse moves over the bar of one gym, we can see colors of the other gyms adjusting automatically. In the example below (Figure 12) the mouse is over Islington Town. As you can instantly see this gym (dark blue) and all other gyms (grey for larger gyms, light blue for smaller ones) in the proximity, you can quickly make comparisons. Most of the other gyms have fewer members than Islington Town.

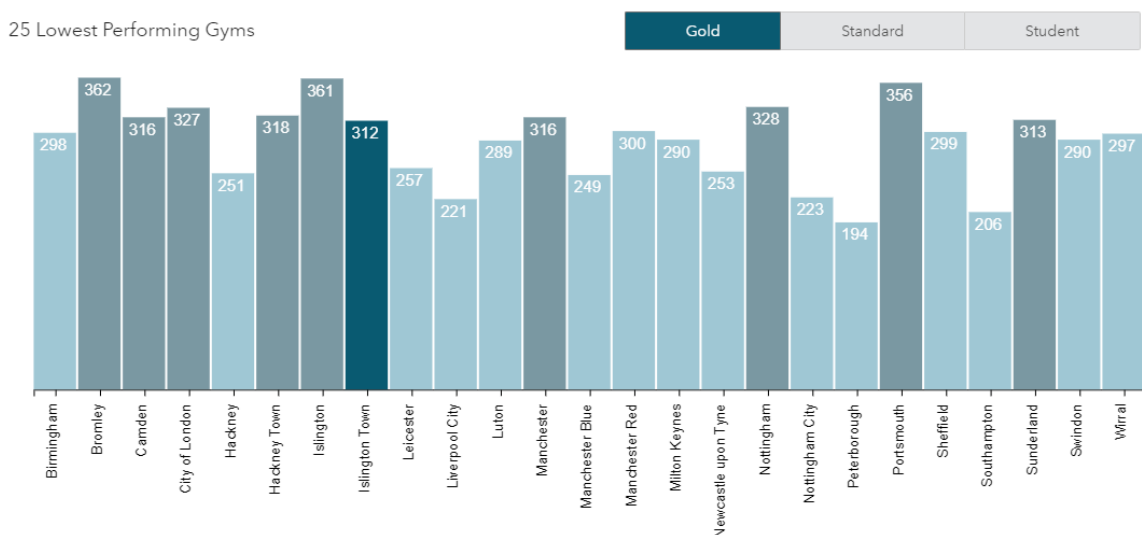


Figure 12. D3.js Bar Chart in SAS Visual Analytics

The Data-Driven Content object in SAS Visual Analytics does require knowledge of JavaScript but will enable report designers to leverage the power of third-party visualizations without leaving SAS Visual Analytics.

CONCLUSION

This paper has discussed three methods that deliver increasing amounts of functionality to change the appearance of visualizations produced by SAS Visual Analytics. The first was the simplest approach and should be every SAS Visual Analytics user's first step to personalize the default output of SAS Visual Analytics.

To introduce new visuals, two further techniques were introduced. The first is creating a custom graph inside the SAS Graph Builder application. The advantages of this method are the GUI interface available to create graphs that have the same look and feel to the default SAS Visual Analytics graphs and is a low complexity solution for combining existing graphs.

The second technique is creating a new graph using a JavaScript framework and importing the graph to your report. This method should be **used if the SAS Graph Builder can't provide** the required functionality. It will take more time to develop but it rewards the user with endless opportunities to design new graphs.

Overall, this paper has added techniques to your toolbox so that next time you need to develop a report you can implement a wider range of visualizations.

REFERENCES

SAS INSTITUTE INC. "Programming Considerations for Data-Driven Visualizations." in SAS Visual Analytics 8.3: Reference. June 2019. Available at <https://documentation.sas.com/?docsetId=varef&docsetTarget=n109mqtyl6quiun1mwfgtcn2s68b.htm&docsetVersion=8.3&locale=en>

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Lucy Smith
Amadeus Software Ltd
Lucy.smith@amadeus.co.uk
www.amadeus.co.uk