Paper SAS517-2017

Nine Best Practices for Big Data Dashboards Using SAS® Visual Analytics

Elena Snavely, SAS Institute Inc., Cary, NC

ABSTRACT

Creating your first suite of reports using SAS® Visual Analytics is like being a kid in a candy store—with so many options for data visualization, it is difficult to know where to start. Having a plan for implementation can save you a lot of time in development and beyond, especially when you are wrangling big data. This paper helps you make sure that you are parallelizing work (where possible), maximizing your data insights, and creating a polished end product. There are guidelines to common questions, such as “How many objects are ‘too many’?” or “When should I use multiple tabs versus report linking?” to start any data visualizer off on the right foot.

INTRODUCTION

A good dashboard can be created with SAS Visual Analytics in minutes, but a great dashboard requires some extra thought and effort. This paper aims to give you a quick starting guide for successful dashboard creation and user adoption. The **Nuts and Bolts** section is an overview of what is going on under the hood of SAS Visual Analytics – and offers some advice on how to speed up performance, make the data more reusable, and ultimately create a robust data model that includes agreed-upon measures from your users. The **Art of the Visual** section is a brief summary of the easiest ways to make your visualizations more interactive, including the types of linking and hierarchies that are available within SAS Visual Analytics. The **Repeat Customers** section offers visual best practices to help guarantee that users will quickly adopt, leverage, and want more of your Big Data Dashboards.

NUTS AND BOLTS

Understand Your Queries

To understand how to best design and optimize Big Data Dashboards using SAS Visual Analytics, it is important to have a working knowledge of what is going on behind the scenes. When a report is first opened in Visual Analytics, every object (such as a map, graph, or other data visualization) is sending a query to the underlying data in SAS® LASR™. For example, in a report with four objects, four separate queries are sent to retrieve the data to populate the visualizations. However, in most dashboards, the number of queries being sent does not stop there. Each object in a container, as well as every report and/or section prompt, is sending a query back to LASR. In the previous example, if one of the 4 objects in your report is actually a stacked container with 3 objects inside it, you are now sending 7 (as opposed to 4) queries back to LASR. If we also have a global prompt on gender, age, and ZIP code, we now have to 10 queries being sent back to LASR.

Unfortunately, this is not just a fun arithmetic problem. In fact, your browser limits the number of queries that can be sent back to LASR at a single time. Table 1 shows the number of queries that can be sent to LASR via the browser (as of December 2016).

Browser	Number of Queries
Chrome 32	6
Firefox 26	6
IE 9	6
IE 10	8
IE 11	13

Browser	**Number of Queries**
Safari 7.0.1	6

Table 1: Number of Queries per Browser[1]

In essence, reports built with SAS Visual Analytics are constrained by elements outside the control of SAS. When you open a report, a package of up to 13 or 8 or 6 queries (depending on your browser) is sent to LASR and then sent back to the report. Before a second package of queries can be sent, the entire first package must be completed. Thus, if your report has 11 queries and you are using IE 11, there is only one package being sent. However, in Firefox, there are two packages of queries – one with 6 and one with 5 – sent to LASR. There is no way to control which query goes with each package, and each completion depends on the slowest query finishing. Even in the world of in-memory analytics, those seconds can add up.

Monitor Your Queries

A quick and easy way to understand how long each individual query takes is found in the performance log. The steps to open the performance log appear below, along with sample output.

1. Open the report in SAS® Visual Analytics Designer.
2. Click the **Properties** tab.
3. Open the performance log via Ctrl-Alt-p.
4. Refresh the report by selecting **File** –> **Refresh Report**.

Output 1 shows an example of the output from your performance log.

```
Processed BIRD
  Time: 253 ms.
Query time for 'Member Zipcode'.
  Time: 756 ms.
Renderer complete for 'Member Zipcode'.
  Time: 764 ms.
Query time for 'Treemap 1'.
  Time: 760 ms.
Query time for 'Geographic'.
  Time: 1170 ms.
Query time for 'Demographics'.
  Time: 1540 ms.
Renderer complete for 'Treemap 1'.
  Time: 1931 ms.
Renderer complete for 'Geographic'.
  Time: 1941 ms.
Renderer complete for 'Demographics'.
  Time: 1481 ms.
Render complete for Section 1
  Time: 10282 ms.
Time taken since report was requested to be opened
  Time: 13470 ms.
```

Output 1. Sample Performance Log for SAS Visual Analytics

For the purposes of understanding each object's query time, focus only on the lines beginning with *Query time*. In the example above, the *Demographics* object's query took the most time, at 1540 milliseconds (1.54 seconds). If you identify a specific query that is taking much longer than others to complete, you can keep the performance log open while you triage the situation. Simply go to **File** –> **Refresh Report** to

[1] http://www.browserscope.org/?category=network

refresh the report after the updates to your report have been made, and your performance log will automatically update with the new times.

More in-depth performance monitoring is available with administrator access. In particular, the Process Monitor tab found in the SAS Visual Analytics Administrator has a huge amount of real-time process information. Refer to the *SAS Visual Analytics: Administration Guide* for details.

Pre-Cook the Data

Users often leverage SAS Visual Analytics to get to answers faster than their typical ad hoc queries to an underlying database. Examples of questions asked in the health care field are “How many patients were readmitted on a given day”, “Which county has the highest cost per patient?”, or “Which condition is driving this cohort’s expenses?”

Some of these questions might have more than one right answer. For example, how are you defining a readmission? Does cost per patient include drug costs? Are costs based on the date of service or date of payment? How are you defining a condition?

Previously, when relying on direct database queries, your answer might change, depending on who is writing the query. By taking a minute to mindfully design and construct not only your visualizations, but the data underlying them, you can achieve consistent, quick answers to those questions. Work with your team to develop flags for important, but defined, elements, and reach consensus on how best to calculate any field which might be up to interpretation. After those flags and established calculations are in place, the potential analysis increases exponentially.

SAS Visual Analytics has a number of ways for users to compute fields on the fly, ways that are not found in their source data. Examples of these calculated items appear in Table 2.

Calculated Item	**Example**
Numeric (simple)	‘x/y’ or ‘x-y’
Numeric (advanced)	‘Floor’ or ‘Root’
Comparison	‘NotBetweenInclusive’ or ‘x < y’
Boolean	‘IF…ELSE’ or ‘OR’
Date and Time	‘DatePart’ or ‘Hour’
Text (simple)	‘Concatenate’ or ‘Format’
Text (advanced)	‘RemoveWord’ or ‘Substring’

Table 2: Calculated Item Examples

Calculated items are a great way to enhance your data during explorations, and can be used effectively in dashboards. However, when building dashboards with multiple queries on big data, every millisecond can add up. In particular, we recommend avoiding Text (simple) and Text (advanced) calculated items in dashboards. Instead, pre-calculate them in the underlying data source before loading into LASR.

Create a Data Model

In SAS Visual Analytics, each object in a visualization is linked to a single table. You cannot display information from two different data sources in a single object. As discussed above, each report object sends a query to LASR to retrieve the data necessary to render the object. Therefore, there are times when using two or more tables in a dashboard, where one would do, actually saves time. The best practices below are some possible remedies when your report performance needs improvement.

When building a report organically, often adding things like report prompts (called controls) happen on the fly. The following controls as available in SAS Visual Analytics: drop-down lists, lists, button bars, text input fields, and sliders. Controls are easy to add, and SAS Visual Analytics automatically creates and displays the information for these report objects. For example, you might have a column in your data called *ZIP code*. When you drag and drop the *ZIP code* variable from your data set into a drop-down list control, the control finds all of the ZIP codes in your data. Now you can select a ZIP code to use as a report filter in your dashboard.

However, when thinking about report performance, a query on a 20 GB table to find the distinct ZIP codes in a data set for the control will take longer than a query on a 128 KB table that contains only the distinct ZIP codes. In-memory processing is fast, but even milliseconds can add up. When developing a report, use the easy drag-and-drop capabilities from your larger data set to create controls. But when it is time to publish your reports, look at your performance using the performance log outlined in **Monitor Your Queries** above, and determine whether smaller tables for these controls might save some time.

Similarly, as each object in a visualization is linked to a single table, consider whether your source data can be broken up into smaller subsets. For example, if one object needs the demographic information of the population from your data, and another object needs the costs associated with each member, separating the source data from one table to two may save time.

After you have completed the design of your dashboards, consider removing any columns from the underlying data which are not in use. This reduces your table size and keeps the queries running as fast as possible. Always keep the variable names in your data logical and consistent. If a future user needs to update your reports, or use them as a springboard for their own dashboards, careful naming can reduce their reliance on a data dictionary. After your main report development is complete, taking the time to trim the underlying tables necessary for your visualizations might save time in your report performance.

ART OF THE VISUAL

Linking Between Reports

Linking between reports is one of the easiest ways to expand the capabilities and thoroughness of your dashboards. By thoughtfully deciding to which information your users will most likely want to drill down, you can design a single *details* report which can be linked to from multiple *overview* reports.

For example, if **Report Z** contains detail information that users often want to know – for instance, a more specific demographic breakdown of information about health care conditions in an area – then you should design **Report Z** so that it can be linked from multiple high-level dashboards.

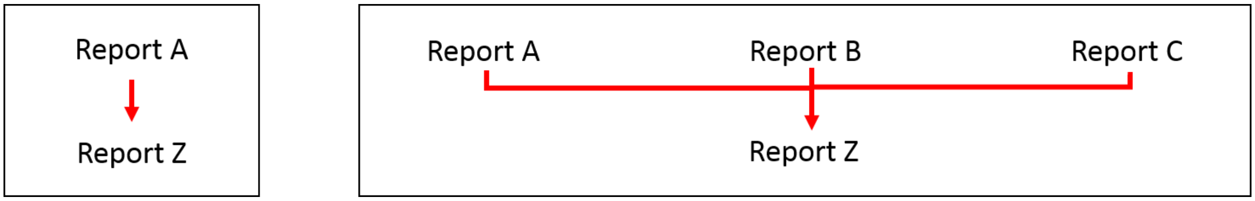


Figure 1: Expanding Report Linkages

In this way, instead of having a single report which goes *A to Z*, you easily have three reports, starting at three distinct locations, but drilling into the same details.

This basic idea can be expanded further. Different objects in a report can link to different detail reports. Thus, any demographic object in a report could link to the demographic detail report – filtered to the specific demographics in question. In the same dashboard, a condition object could link to a condition detail report – filtered to the specific condition in question. The dashboard objects themselves become springboards into a deeper understanding, depending on the needs and interests of users.

Leveraging Multiple Sections

While linking between different reports allows for a more detailed view of a specific area of interest, creating multiple sections under the same global filters allows for a broader view across areas of interest, with very little effort from users.

The best benefit of multiple sections is that you do not have to cram everything into a single page. Sections can be based on different areas of focus or different levels of detail. They can use the same basic layout, or they can highlight different measures using different visualizations.

For example, consider a dashboard created to show drug prices. The dashboard might contain information about trends, geography, providers and members of interest, and so on. Some users might care only about total drug prices, while others care about drug usage, and a third group wants access to both sets of information, but with the ability to subset the data according to specific parameters (dates, demographics, and so on). One way to accommodate these multiple needs is to create a single report with multiple sections that focus on different aspects of interest, and to use global filters to allow users to subset the reports to their particular areas of interest.

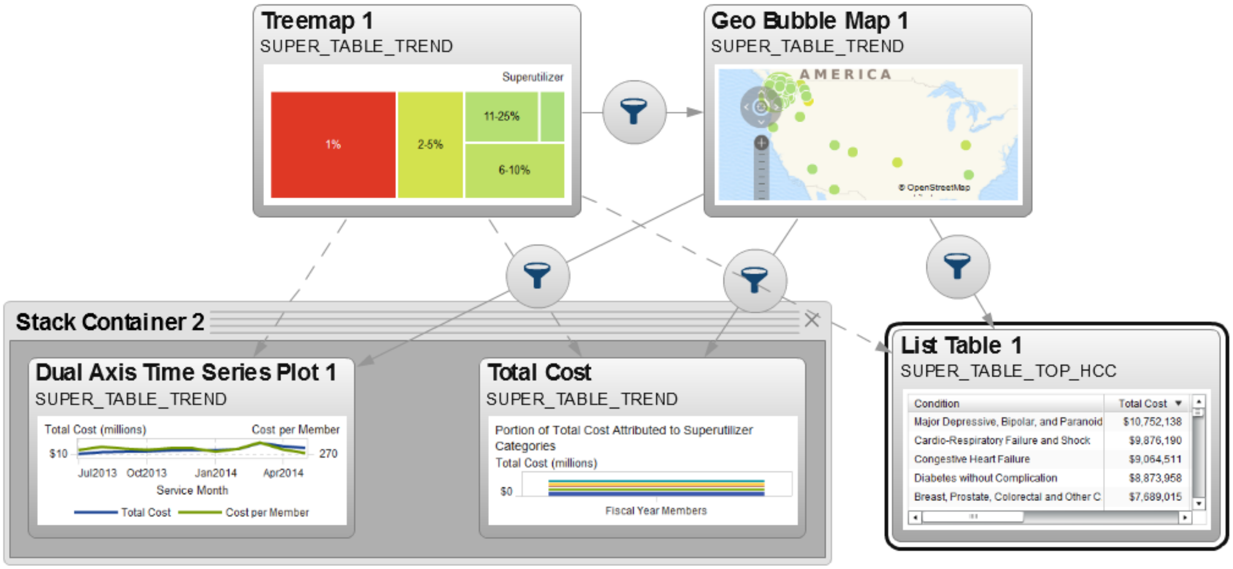
Interactions within a Report

Interactions within a report are very important. They are used to both update information in the report and to draw attention to specific results. There are two types of interactions – filtering (which subsets the returned data) and brushing (which highlights data shared by two or more objects). Filtering is the more intuitive of the two types of interactions in dashboards, and is often used to highlight the interactive features of dashboards.

In SAS Visual Analytics, filter interactions must be set up in a directional manner. An object cannot both filter and be filtered by another object. In other words, only one direction of filtering is allowed. Thus, it is important to keep the filter interactions as consistent as possible.

In Figure 2, the report objects are filtered clockwise, starting from the Treemap1 in the upper left corner. Notice that in order for all of the objects in a stacked container to be mapped to the proper interaction, each separate object must be linked. In this example, Treemap 1 filters both Geo Bubble Map 1, List Table 1, Dual Axis Time Series Plot 1, and Total Cost. Geo Bubble Map 1 filters List Table 1, Dual Axis Time Series Plot 1, and Total Cost. And so on.

After you determine a direction of the filter interactions for one dashboard, consider using that as your template going forward. It is important that the user's experience across reports remains consistent.



Condition	Total Cost ▼
Major Depressive, Bipolar, and Paranoid	$10,752,138
Cardio-Respiratory Failure and Shock	$9,876,190
Congestive Heart Failure	$9,064,511
Diabetes without Complication	$8,873,958
Breast, Prostate, Colorectal and Other C	$7,689,015

Figure 2: Sample Clockwise Filter Interactions

REPEAT CUSTOMERS

Not All Visualizations Are Created Equal

When deciding how best to represent your data, it is important to think about how many dimensions you are trying to represent. Are you trying to show frequency of gender (one dimension)? What about frequency of gender over time (two dimensions)? List tables and Crosstabs always allow you to include as many columns from the data as you wish. However, if using a more visual representation is your goal, it is important to remember that different data visualizations have different advantages and limitations.

For example, a basic bar chart has two data roles: the height and category of the bars. A basic bubble plot has three data roles: X axis, Y axis, and Bubble Size. Some of the additional data roles available for visualizations are: Animation, Data tip values, Group, Lattice columns, Lattice rows, and Color. Table 3 shows the various report object and their associated numbers of roles.

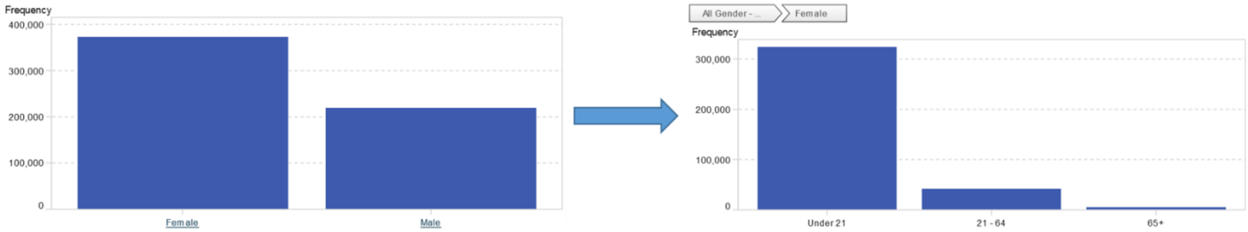
Report Object	Number of Basic Data Roles	Number of Additional Data Roles
Bar Chart	2	5
Waterfall Chart	2	3
Line Charts	2	5
Pie Chart	2	5
Scatter Plot	2	4
Time Series Plot	2	2
Bubble Plot	3	6
Treemap	2	2
Dual axis bar chart	3	4
Dual axis line chart	3	4
Dual axis bar-line chart	3	4
Dual axis time series plot	3	1
Geo bubble map	1	4
Gauge	1	3
Word Cloud	3	0

Table 3: Basic Report Objections and Their Number of Basic and Additional Roles

Choose a visualization that is appropriate for what you are trying to show, and be mindful of which differentiators are important for your users. Do different colors pop more than different sizes? Always be mindful of accessibility when determining whether contrasting colors (for example: red/green) are visible to your users. For more information about the different types of report objects and their potential data roles, see the **Working with Data Role Assignments** section of the *SAS Visual Analytics: User Guide*.

A common mistake of new data visualizers is to confuse Line Chart and Time Series Plots. On the surface, both look the same. However, if you enter a date, time, or datetime data item into the category role of the Line Chart, you can end up with a sort order that is not timely. Instead, be sure to use the Time Series Plot for any line graphs involving time.

Finally, hierarchies are always a great way to add detail to the information available, without adding complexity to the visualization itself. Hierarchies add drill-down information to a report. For example, you could have a hierarchy from Member Gender to Member Age. The initial bar chart shows frequency by gender, but by double-clicking on the *Female* bar, users drill-down into the data and see the breakdown by age categories for the female population.



Display 1. Drill from Initial Gender Bar Chart into the *Female* Category

Another benefit of hierarchies is that they add a layer of interaction, which can increase the use and adoption of your reports. Hierarchies are not available in all report objects, and there is a limit to a maximum of two hierarchies per object. For more information about hierarchies, see **Working with Hierarchies in a Report** section of the *SAS Visual Analytics: User Guide*.

Be Consistent

The single best way to ensure use and adoption of your dashboards is to be consistent. If bar charts are how you typically show the frequency of gender, then do not switch to pie charts to show frequency of gender in a different dashboard. If your gender graphs typically link to a detailed demographics report, do not forget to link all of the gender graphs in all of your dashboards to that detail report. If your filter interactions start in the upper left corner and go clockwise, be consistent with both the starting location of interactions and the direction in other reports.

Take the time to think about your dashboard, how your users are accustomed to viewing information, and how you have displayed information in the past. By following the above best practices, you will soon be making better visualizations, faster.

CONCLUSION

When creating dashboards with SAS Visual Analytics, the possibilities are almost limitless. To create dashboards that will be quickly adopted, you must take some time to properly design your reports. By using the information here, you have the foundation to establish a data model, use linking appropriately, mindfully choose visualizations, as well as know some quick tips for troubleshooting if performance needs to be tuned.

ACKNOWLEDGMENTS

These best practices were first outlined in 2016 with the help and hard work of Matthew Schirmer, Principal Technical Consultant at SAS, and Eric Whitney, Implementation Project Manager at SAS. Without them, this paper would never have been possible.

RECOMMENDED READING

- *SAS® Visual Analytics: User Guide.*
- *SAS® Visual Analytics: Administration Guide*

CONTACT INFORMATION

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

Elena Snavely
SAS Institute Inc.
100 SAS Campus Drive
Cary, NC 27513
(919) 531-2057
Elena.snavely@sas.com
http://www.sas.com