Paper 1010-2017

SAS® Visual Analytics Tricks We Learned from Reading Hundreds of SAS® Community Posts

Tricia Aanderud, Ryan Kumpfmiller; Zencos Consulting
Rob Collum, SAS Institute

ABSTRACT

After you know the basics of SAS Visual Analytics, you realize there are some situations that require unique strategies. Sometimes tables are not structured right or become too large for the environment. Maybe creating the right custom calculation for a dashboard can be confusing. Geospatial data is hard to work with if you haven't ever used it before. We looked through 100s of SAS Communities posts for the most common questions. These solutions (and a few extras) were extracted from the newly released *Introduction to SAS Visual Analytics* book.

INTRODUCTION

Our goal in writing the *Introduction to SAS Visual Analytics* book was to create a really practical and useful book for users. As part of the research for the book, we read hundreds of posts in the SAS Communities: SAS Visual Analytics section. This was a great exercise because we could confirm some of the sticking points that we know users have and pick up some tips to emphasize in the book as well. This paper combines our favorite tips from the book and some other ones that we think are worth sharing.

Note: All examples were done using SAS Visual Analytics version 7.3

WHAT IS SAS COMMUNITIES?

SAS Communities is an open forum on the SAS website, https://communities.sas.com, which connects SAS users all over the world. With over a hundred thousand members, users can post questions about challenges they are currently having working with SAS products or provide answers to other users who are looking for advice. The site is structured with communities around all the SAS products so it's easy to find the topics you're looking for.

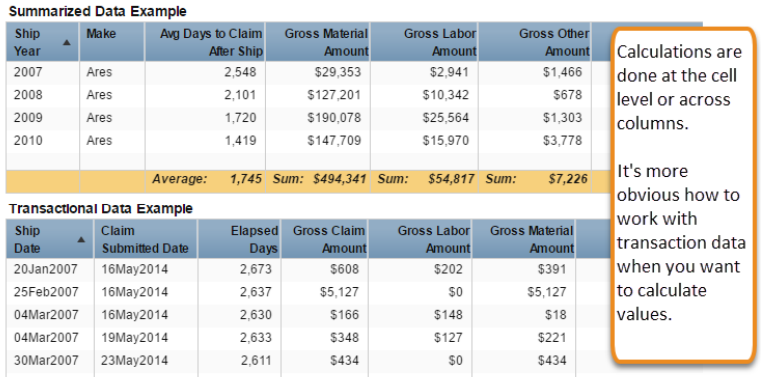
CREATING DATA ITEMS

SAS Visual Analytics is meant to consume any kind of data. This includes SAS datasets, spreadsheets, database tables, and even social media data. These data sources can come in all different sizes and forms. To get the insights, there's sometimes some initial work that needs to be done on that data set before introducing objects and visualizations. Here's some tips to help.

WHAT SHAPE IS YOUR DATA?

Some new users don't understand that the shape of the data matters when working with SAS Visual Analytics. If your data is summarized instead of transactional, you might find creating reports and data items more challenging. In the following figure, you can see the difference in the data shapes.

Knowing the shape of your data is important when you want to create calculations. With simple calculations you can go across the columns to get a new value. With Aggregations you can go across and down. With a derived measure you can create date based calculations.



Summarized Data Example

Ship Year	Make	Avg Days to Claim After Ship	Gross Material Amount	Gross Labor Amount	Gross Other Amount
2007	Ares	2,548	$29,353	$2,941	$1,466
2008	Ares	2,101	$127,201	$10,342	$678
2009	Ares	1,720	$190,078	$25,564	$1,303
2010	Ares	1,419	$147,709	$15,970	$3,778
		Average: 1,745	Sum: $494,341	Sum: $54,817	Sum: $7,226

Transactional Data Example

Ship Date	Claim Submitted Date	Elapsed Days	Gross Claim Amount	Gross Labor Amount	Gross Material Amount
20Jan2007	16May2014	2,673	$608	$202	$391
25Feb2007	16May2014	2,637	$5,127	$0	$5,127
04Mar2007	16May2014	2,630	$166	$148	$18
04Mar2007	19May2014	2,633	$348	$127	$221
30Mar2007	23May2014	2,611	$434	$0	$434

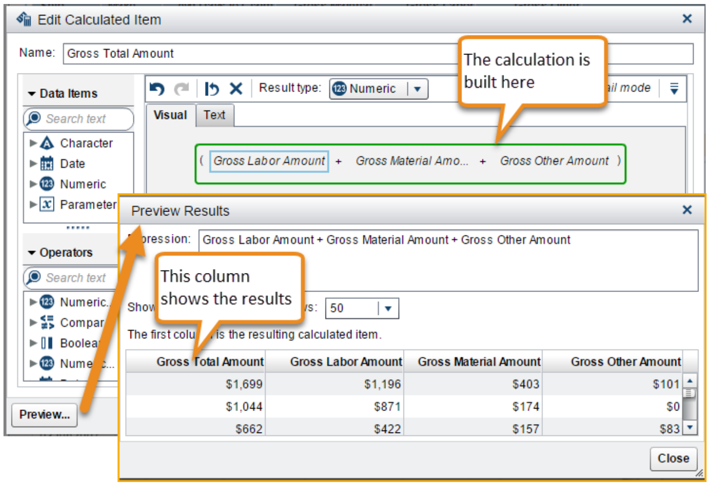
Calculations are done at the cell level or across columns.

It's more obvious how to work with transaction data when you want to calculate values.

CREATING CALCULATIONS FROM DATA ITEMS

You can create new data items from existing data items using a calculation. Calculations allow you to create simple formulas from numeric data items or apply functions to character data items. To add a calculation, select Add Calculation from the Data drop-down menu.

In the example to the right, the Gross Total Amount data item was created based on the existing data items. This calculation could be based on a summarized or transaction dataset. The calculation goes across the columns and provides the answer for each row.



Name: Gross Total Amount

The calculation is built here

(Gross Labor Amount + Gross Material Amo... + Gross Other Amount)

pression: Gross Labor Amount + Gross Material Amount + Gross Other Amount

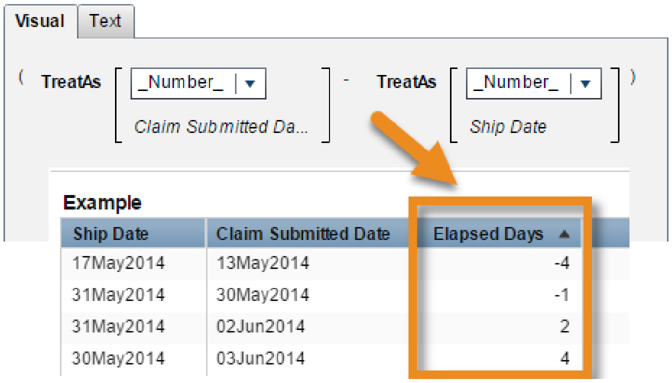
This column shows the results

Show ...s: 50

The first colu... is the resulting calculated item.

Gross Total Amount	Gross Labor Amount	Gross Material Amount	Gross Other Amount
$1,699	$1,196	$403	$101
$1,044	$871	$174	$0
$662	$422	$157	$83

Getting the Difference in Dates



(TreatAs [_Number_ Claim Submitted Da...] - TreatAs [_Number_ Ship Date])

Example

Ship Date	Claim Submitted Date	Elapsed Days
17May2014	13May2014	-4
31May2014	30May2014	-1
31May2014	02Jun2014	2
30May2014	03Jun2014	4

A trickier question is how do I calculate the difference in two dates. In the SAS language, dates are treated like a numeric value but in SAS Visual Analytics the date values are closer to character values. The trick is to convert the date to a number before doing your calculation using the TreatAs() function.

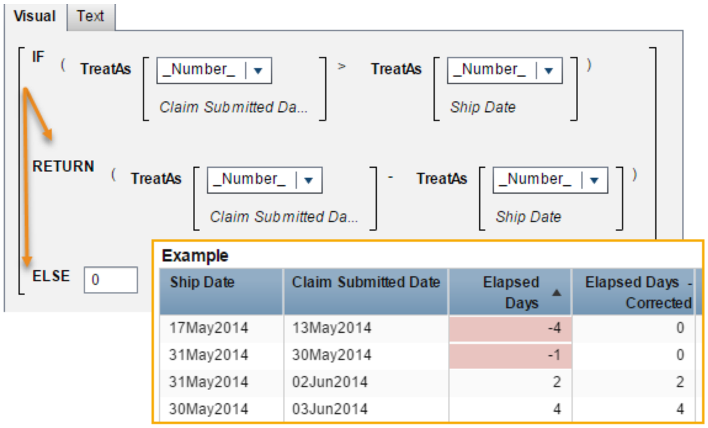
In this example we calculate the days between product shipment and date a claim was submitted. The data items must be on the same row for the calculation to work. The TreatAs function is applied to each date before the calculation.

Tip! Add the Datepart() function if you have a datetime value.

Adding Logic Statements to Transform Value

You can use IF/THEN logic to create a data item. In the preceding example the calculation returned some negative numbers. The data has an error since it unlikely a repair was made to an unshipped item. In the example, we add the IF/THEN function to test if the result is greater than 0 and then do the calculation. If it fails, we set the value to 0. Place your most likely scenario first in the IF/THEN logic.

You can use IF/THEN logic with parameters and other calculations.



IF (TreatAs [_Number_ Claim Submitted Da...] > TreatAs [_Number_ Ship Date])

RETURN (TreatAs [_Number_ Claim Submitted Da...] - TreatAs [_Number_ Ship Date])

ELSE 0

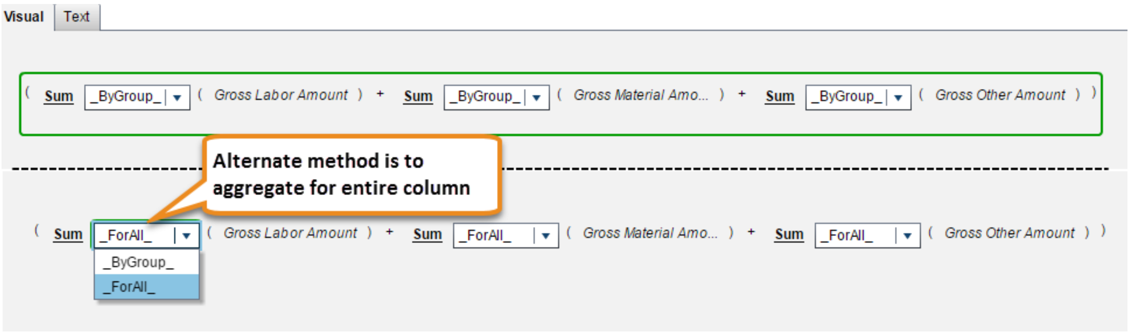
Example

Ship Date	Claim Submitted Date	Elapsed Days	Elapsed Days - Corrected
17May2014	13May2014	-4	0
31May2014	30May2014	-1	0
31May2014	02Jun2014	2	2
30May2014	03Jun2014	4	4

Creating Aggregated Measures

Aggregated measures are difficult to understand when you first start. These data items are similar to the calculated items in that you can create calculations and logical expression results. It is often not obvious how the two data items are different. There is an easy way to think of each one. Calculated items go across the rows while aggregated measures go across and then down the columns.

In a previous example we created the Gross Total calculation based on three data items. In this example the same calculation is created as an Aggregated Measure. It has the aggregations applied so it appears differently. Note that you can change the aggregations to average, maximum, minimum, and so on.

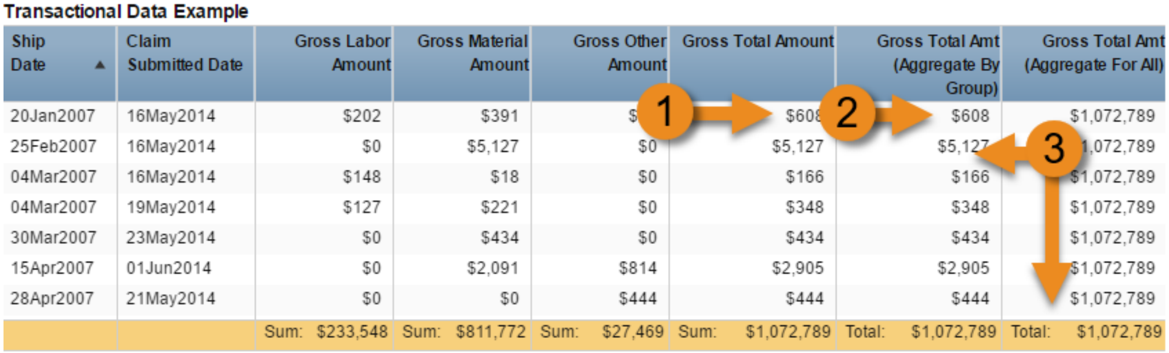


(Sum _ByGroup_ (Gross Labor Amount) + Sum _ByGroup_ (Gross Material Amo...) + Sum _ByGroup_ (Gross Other Amount))

Alternate method is to aggregate for entire column

(Sum _ForAll_ (Gross Labor Amount) + Sum _ForAll_ (Gross Material Amo...) + Sum _ForAll_ (Gross Other Amount))

In the following figure, notice the difference in how each calculation appears. Our first calculation goes across the columns. The second calculation is set up as a _ByGroup_ and goes across the columns. The third aggregation is a _ForAll_ and goes across and down the columns.



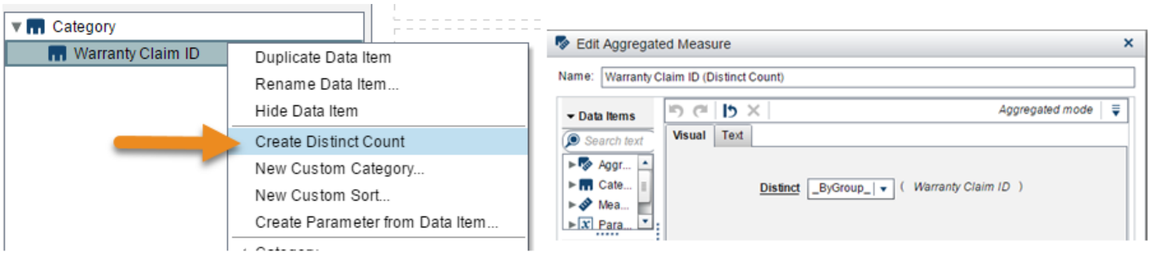
Transactional Data Example

Ship Date ▲	Claim Submitted Date	Gross Labor Amount	Gross Material Amount	Gross Other Amount	Gross Total Amount	Gross Total Amt (Aggregate By Group)	Gross Total Amt (Aggregate For All)
20Jan2007	16May2014	$202	$391	$	$608	$608	$1,072,789
25Feb2007	16May2014	$0	$5,127	$0	$5,127	$5,127	1,072,789
04Mar2007	16May2014	$148	$18	$0	$166	$166	$1,072,789
04Mar2007	19May2014	$127	$221	$0	$348	$348	$1,072,789
30Mar2007	23May2014	$0	$434	$0	$434	$434	$1,072,789
15Apr2007	01Jun2014	$0	$2,091	$814	$2,905	$2,905	$1,072,789
28Apr2007	21May2014	$0	$0	$444	$444	$444	$1,072,789
		Sum: $233,548	Sum: $811,772	Sum: $27,469	Sum: $1,072,789	Total: $1,072,789	Total: $1,072,789

There’s not a hard and fast rule for when to use an aggregation over a calculation. It usually depends on where the calculation is being used. Aggregations are more flexible and change as your data object changes. What you might notice about the aggregation is that we cannot compare row values. For that we need a derived measure.

Other Aggregation Examples

You can also use aggregations to get a unique count of categorical data items. For instance, if the data contains multiple Warranty Claim IDs based on each stage of the process. You can use the Distinct Count feature to get the unique number of warranties in the dataset. This value can be used in other aggregation calculations.



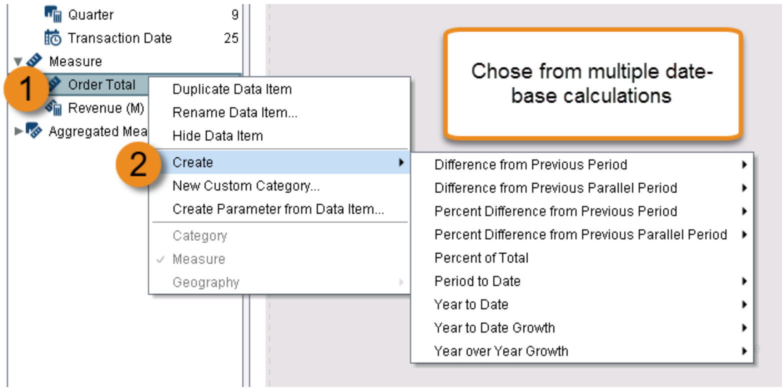
Name: Warranty Claim ID (Distinct Count)

Distinct _ByGroup_ (Warranty Claim ID)

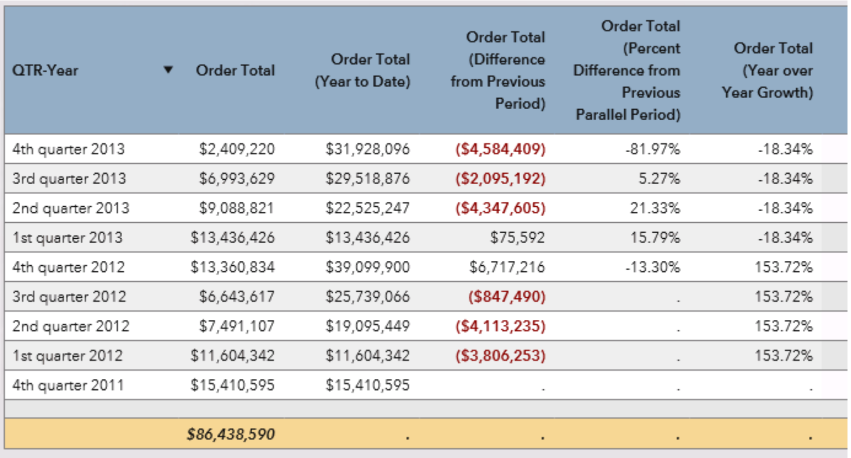
Creating Derived Items

Derived items are created from an existing measure. Right-click a measure and select Create and choose from one of date-based aggregated measures. After the measure is created, it is added to the Aggregated Measure list

When you create any calculated data items, you should test it prior to use. The derived measures can be especially tricky since it's not always obvious what is being calculated. In the following figure, four example derived items were created.



Chose from multiple date-base calculations



QTR-Year ▼	Order Total	Order Total (Year to Date)	Order Total (Difference from Previous Period)	Order Total (Percent Difference from Previous Parallel Period)	Order Total (Year over Year Growth)
4th quarter 2013	$2,409,220	$31,928,096	($4,584,409)	-81.97%	-18.34%
3rd quarter 2013	$6,993,629	$29,518,876	($2,095,192)	5.27%	-18.34%
2nd quarter 2013	$9,088,821	$22,525,247	($4,347,605)	21.33%	-18.34%
1st quarter 2013	$13,436,426	$13,436,426	$75,592	15.79%	-18.34%
4th quarter 2012	$13,360,834	$39,099,900	$6,717,216	-13.30%	153.72%
3rd quarter 2012	$6,643,617	$25,739,066	($847,490)	.	153.72%
2nd quarter 2012	$7,491,107	$19,095,449	($4,113,235)	.	153.72%
1st quarter 2012	$11,604,342	$11,604,342	($3,806,253)	.	153.72%
4th quarter 2011	$15,410,595	$15,410,595	.	.	.
	$86,438,590	.	.	.	.

- **Order Total (Year to Date)** shows the cumulative amount for each year. It resets in first quarter.
- **Order Total (Difference from Previous Period)** is the difference in the previous period and this quarter.
- **Order Total (Percent Difference from Previous Parallel Period)** is the percentage difference in the same quarter of the previous year.
- **Order Total (Year over Year Growth)** is the percentage difference in the same quarter of the previous year.

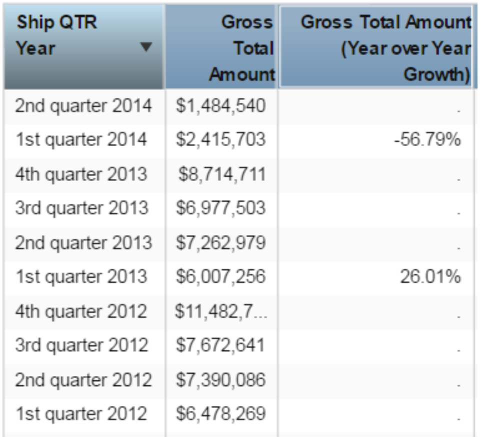
Each derived measure was based on the Order Total column. Notice that the total row does not contain any values for the derived measures.

Understanding the Parallel Period Derived Item

Let's look at the parallel period derived item based on the Gross Total Amount measure. Parallel periods can be a bit tricky to understand at first since there's a lot of options used in the formula that can be manipulated. In this example, we created a *Year Over Year Growth* derived item.

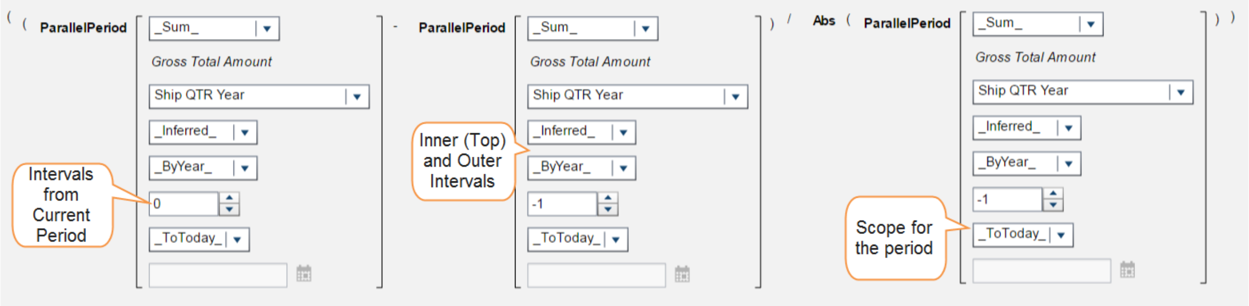
By default, you only see results for the first quarter of each year when the derived item is grouped by quarters instead of year. The percentages appear incorrect when you look at total year-over-year amounts or quarterly results.

This can be adjusted so we see the results for each quarter.



Ship QTR Year ▼	Gross Total Amount	Gross Total Amount (Year over Year Growth)
2nd quarter 2014	$1,484,540	.
1st quarter 2014	$2,415,703	-56.79%
4th quarter 2013	$8,714,711	.
3rd quarter 2013	$6,977,503	.
2nd quarter 2013	$7,262,979	.
1st quarter 2013	$6,007,256	26.01%
4th quarter 2012	$11,482,7...	.
3rd quarter 2012	$7,672,641	.
2nd quarter 2012	$7,390,086	.
1st quarter 2012	$6,478,269	.

When you edit the aggregated data item, here is what the formula looks like:



((ParallelPeriod [_Sum_ | Gross Total Amount | Ship QTR Year | _Inferred_ | _ByYear_ | 0 | _ToToday_] - ParallelPeriod [_Sum_ | Gross Total Amount | Ship QTR Year | _Inferred_ | _ByYear_ | -1 | _ToToday_]) / Abs (ParallelPeriod [_Sum_ | Gross Total Amount | Ship QTR Year | _Inferred_ | _ByYear_ | -1 | _ToToday_]))

Intervals from Current Period

Inner (Top) and Outer Intervals

Scope for the period

You can think of the formula in more simple terms like this:

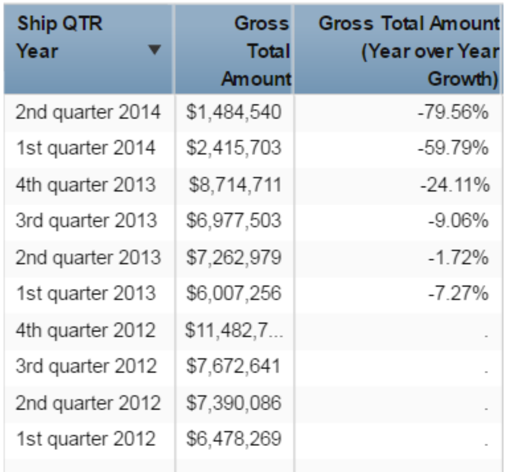
(Period this year – Period last year) / Period last year

This formula is computing the difference between periods as you would for any amounts by subtracting the previous amount from the current amount and then dividing by the previous amount. The different periods can be determined from a field that counts the intervals from the current period. This is displayed in first section in the image above. The value 0 looks at the period of the current date that the row is in. The other parallel period functions have a -1 so that they look one period back based on the selections that you have made in the inner and outer intervals, which are the fields pointed out in the second Parallel Period function above.

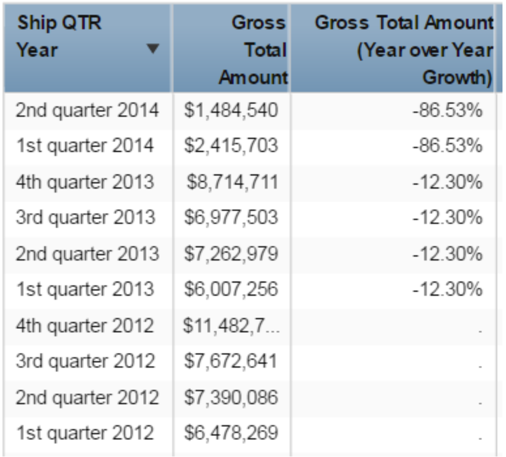
We will change those in a few steps, but first we have to figure out why our percentages do not make sense, which is due to the scope of the period (shown in the third function of the figure). The scope tells the formula how much of a period to aggregate. By using the **_ToToday_** option we are telling the system to run the calculations for today's date (February 27th).

This is why we are only getting results for the first quarters of each year and why those numbers don't even match when looking at the first quarters of the years. The formula is only taking the Gross Total Amount for transactions from January 1st to February 27th and not the whole first quarter. This option is what you would want to use if you had a dashboard for the current year and were tracking data as it got updated daily, weekly, etc.

When we change the option to **_Full_** then the formula takes the full quarter of each date and we get the results shown on the right. The quarters for 2012 are still missing because there is no previous data to compare to, but now we have the year-over-year totals comparing every quarter to the quarter from the previous year.



Ship QTR Year ▼	Gross Total Amount	Gross Total Amount (Year over Year Growth)
2nd quarter 2014	$1,484,540	-79.56%
1st quarter 2014	$2,415,703	-59.79%
4th quarter 2013	$8,714,711	-24.11%
3rd quarter 2013	$6,977,503	-9.06%
2nd quarter 2013	$7,262,979	-1.72%
1st quarter 2013	$6,007,256	-7.27%
4th quarter 2012	$11,482,7...	.
3rd quarter 2012	$7,672,641	.
2nd quarter 2012	$7,390,086	.
1st quarter 2012	$6,478,269	.



Ship QTR Year ▼	Gross Total Amount	Gross Total Amount (Year over Year Growth)
2nd quarter 2014	$1,484,540	-86.53%
1st quarter 2014	$2,415,703	-86.53%
4th quarter 2013	$8,714,711	-12.30%
3rd quarter 2013	$6,977,503	-12.30%
2nd quarter 2013	$7,262,979	-12.30%
1st quarter 2013	$6,007,256	-12.30%
4th quarter 2012	$11,482,7...	.
3rd quarter 2012	$7,672,641	.
2nd quarter 2012	$7,390,086	.
1st quarter 2012	$6,478,269	.

Now our calculation is for the year-over-year growth, what if we just wanted to look at the years?

To change what amounts are calculated we can change the intervals from the inner and outer fields that were shown in the full formula. Right now in our formula, our inner interval is _**Inferred**_ and our outer is _**ByYear**_. When you use the _**Inferred**_ option, the formula will look at the row date used for the derived measure and determine what should be used. Since we are using Ship QTR, the inner interval resolves to quarter. When we change that to _**ByYear**_ we get the results on the left. Now we have our year over year growth based just on the year.

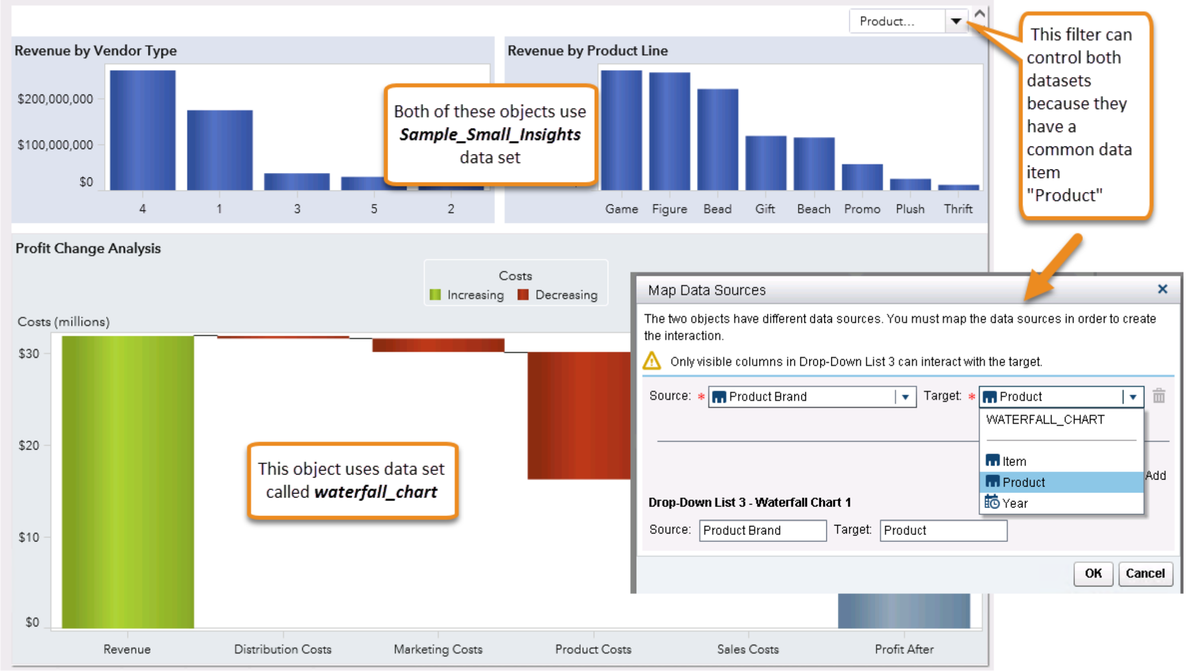
WORKING WITH DATA SOURCES

Reporting is nothing without its data sources. SAS Visual Analytics has some underrated capabilities when it comes to handling data. Below are examples of how to handle situations such as needing multiple datasets in a report, downloading the source data, and swapping out a dataset for a new one.

USING MULTIPLE DATA SOURCES IN REPORT

You can use multiple data sources in your report. Just by loading in the extra tables. Each object can have its own data source. In the following figure, two bar charts use the SAMPLE_SMALL_INSIGHTS data set. The waterfall chart uses the WATERFALL_CHART dataset. When the filter control at the top was added, it also uses the SAMPLE_SMALL_INSIGHTS dataset. We want the filter to control all objects.

If the data sources have a common data item, you can map the data sources. In this case, the datasets have Product (or Product Brand) as a common data item. The data items do not need to have the same name. After adding your data objects and control, right-click the control and select Edit Data Source Mapping. In the Source field, select the data items for the Source and the Target that should be mapped.



Both of these objects use ***Sample_Small_Insights*** data set

This filter can control both datasets because they have a common data item "Product"

This object uses data set called ***waterfall_chart***

Map Data Sources

The two objects have different data sources. You must map the data sources in order to create the interaction.

⚠ Only visible columns in Drop-Down List 3 can interact with the target.

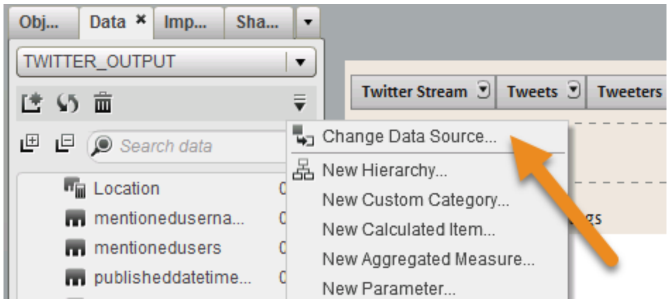
Source: * Product Brand Target: * Product

Drop-Down List 3 - Waterfall Chart 1

Source: Product Brand Target: Product

CHANGING OUT A DATASET

When you are developing a report, you might discover that your data source needs additional data items or you want to rename your dataset to match the report. This can be done but there are a few issues you can side-step for a successful transition.



This technique is most successful when your dataset matches your existing dataset exactly. However, if your dataset is altered, for instance you renamed a data item or changed a data item from character or numeric – it might cause some issues.

If the altered data item is not used in a calculation or data object, it won't matter. Otherwise, there is going to be some trouble. In some cases, SAS Visual Analytics can recognize the issue and make the repair. When you open the report in the Designer, you are prompted to fix the items. Depending on what changed, this action can work. However, if the data has had a more dramatic change, you might need to edit the underlying XML file. Contact SAS Technical Support for assistance with this step.

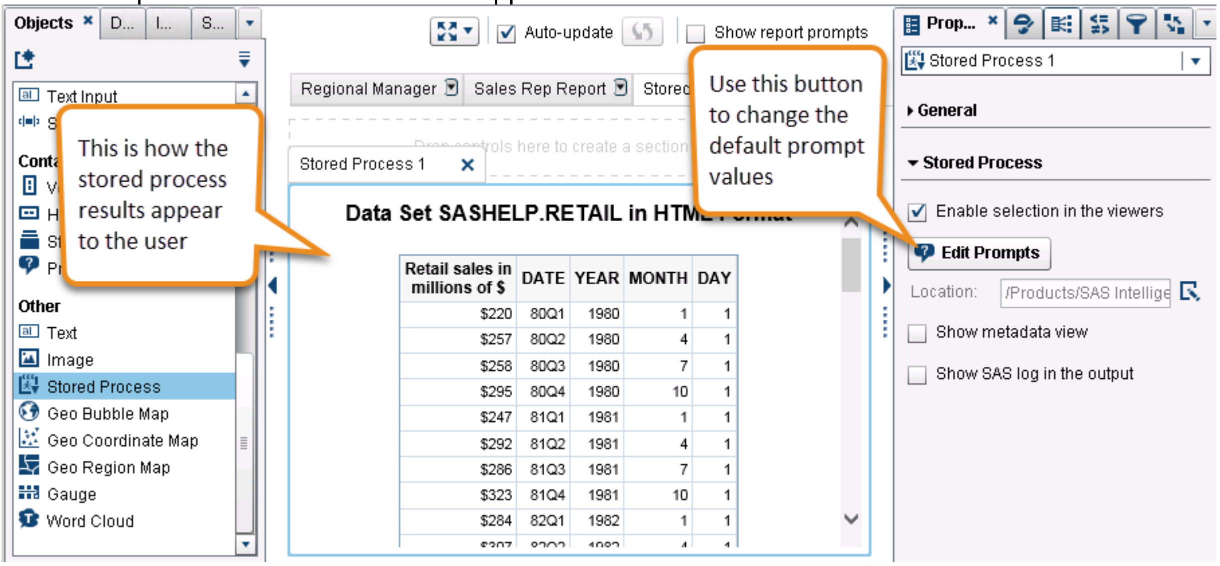
EMBEDDING A STORED PROCESS

From SAS Visual Analytics, you can embed a stored process in your report or link to a stored process from a data object. When embedding a stored process in your report, you can take advantage of existing stored processes or add other SAS-based reports.

SAS Visual Analytics has a stored process object that allows you to place the stored process within the report. However, the stored process does not interact with the other elements of the report. For instance, you cannot use the report or section filters on the stored process or use the prompt choices to interact with other objects. In the following example, a sample stored process is inserted in the section.

1. Create a new section/page in your report.
2. Drag the **Stored Process** object to the work area.
3. When prompted, navigate to the **SAS Folders > Products > SAS Intelligence Platform > Sample** folder and select the **Sample: Multiple Output Formats** stored process from its metadata location.

   The stored process runs and the results appear in the section.



This is how the stored process results appear to the user

Use this button to change the default prompt values

☑ Auto-update

□ Show report prompts

☑ Enable selection in the viewers

Location: /Products/SAS Intellige

□ Show metadata view

□ Show SAS log in the output

Data Set SASHELP.RETAIL in HTML Format

Retail sales in millions of $	DATE	YEAR	MONTH	DAY
$220	80Q1	1980	1	1
$257	80Q2	1980	4	1
$258	80Q3	1980	7	1
$295	80Q4	1980	10	1
$247	81Q1	1981	1	1
$292	81Q2	1981	4	1
$286	81Q3	1981	7	1
$323	81Q4	1981	10	1
$284	82Q1	1982	1	1
$307	82Q2	1982	4	1

WORKING WITH REPORT LAYOUT

While using a report building tool like SAS Visual Analytics, sometimes you may find it difficult to put in the right feature or object for display. However, these types of constraints can usually be worked around with a bit of creativity. Here are some pointers on working with the layout and display of your report.

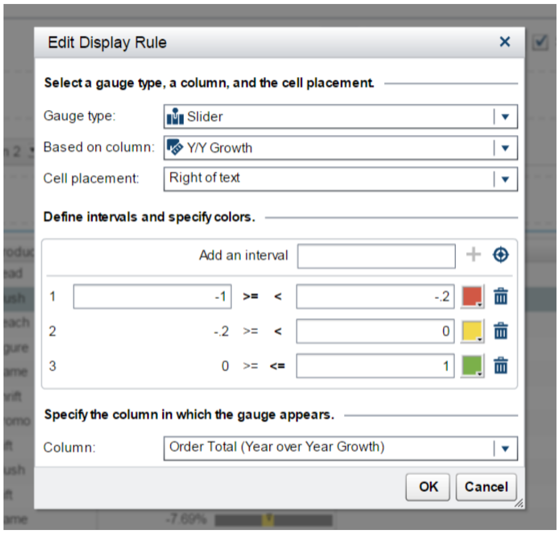
ADDING GAUGES TO A TABLE

Gauges are great to use when building a dashboard, but sometimes they can take up a lot of space that you might want to use for additional objects. One way to get around that is to put them into a list table. Three of the gauge types (bullet, slider, and thermometer + an icon version) can be used as a display rule in a list table. This type of gauge works exactly the same as the object version. When you think about having quick takeaways in your report, this is a very efficient way to do it.

Earlier in this paper we created a derived measure with the Year Over Year Growth formula. Using a data item with that formula, were going to put that into a list table and use a gauge to tell us how were doing.

Add a gauge or display rule to the list table.

1. Right-click the column that contains the measure you want to use with gauge and select Add Display Rule from the pop-up menu. The New Display Rule window appears.
2. Select Gauge to add a new gauge.
3. Select a Gauge type.
4. If your measure does not appear in the Based on column, you can select it from the drop-down list.
5. Decide if you want the icon to appear to the left or right of the column value.
6. Set the display rules to determine the colors. This gauge uses reverse logic.
7. Click Ok to continue.


Edit Display Rule

Select a gauge type, a column, and the cell placement

Gauge type: Slider

Based on column: Y/Y Growth

Cell placement: Right of text

Define intervals and specify colors.

Add an interval

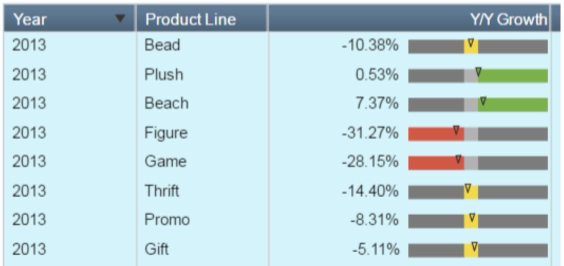
1	-1	>=	<	-2	🟥
2	-2	>=	<	0	🟨
3	0	>=	<=	1	🟩

Specify the column in which the gauge appears.

Column: Order Total (Year over Year Growth)

Review the Results

After completing the display rule in our example, we get the slider gauges to appear next to the Y/Y Growth percentages that came from the derived measure, as shown here on the right. Notice how the slider is the exact same as if it were an object with the color differences and the indicator that tells us exactly where the number falls according to our intervals. This gives us an instant indication on which objects have been struggling compared to the previous year and which ones have surpassed their previous numbers.

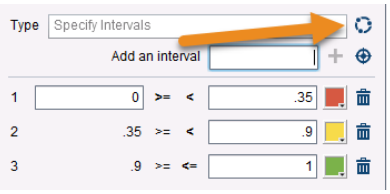

Year	Product Line	Y/Y Growth
2013	Bead	-10.38%
2013	Plush	0.53%
2013	Beach	7.37%
2013	Figure	-31.27%
2013	Game	-28.15%
2013	Thrift	-14.40%
2013	Promo	-8.31%
2013	Gift	-5.11%

USE SHARED DISPLAY RULES

If your dashboard has several gauges all using the same display rules, then create a shared rule. A shared rule allows you to create the display rule once and use it multiple times. In this example, each of these gauges use a 0 to 100% scale where over 90% is the desired goal.

Create Shared Rule

1. Create the display rule. Click the Shared Rule icon.

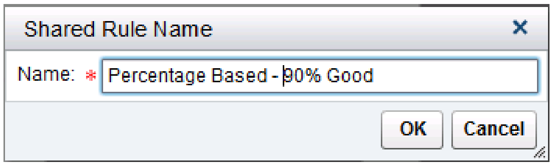


Type: Specify Intervals

Add an interval

1	0	>=	<	.35
2	.35	>=	<	.9
3	.9	>=	<=	1

2. When prompted, name the shared rule. The shared rule is applied to the current gauge automatically.

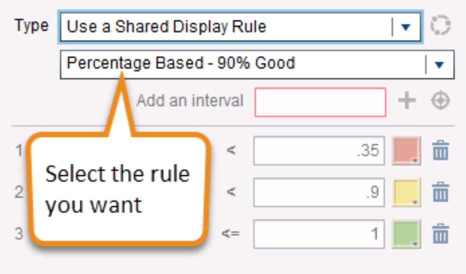


Shared Rule Name

Name: * Percentage Based - 90% Good

Apply to Other Gauges

To apply the shared rule to a gauge, select **Use a Shared Display Rule** choice from the Type field. Then select the desired rule.



Type: Use a Shared Display Rule

Percentage Based - 90% Good

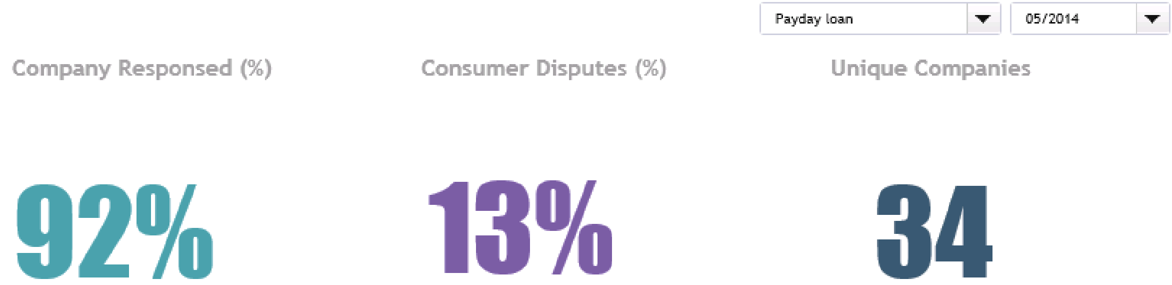
Add an interval

Select the rule you want

1	<	.35
2	<	.9
3	<=	1

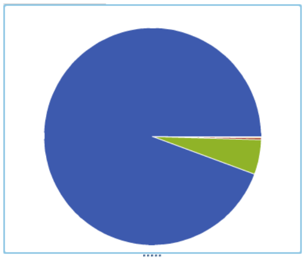
NEED AN INFOGRAPHIC LOOK?

If you are working on a dashboard, sometimes you want to display a single value. Similar to the way infographics look. You could use a single cell in a list table, but here's another method. Use an invisible pie chart.

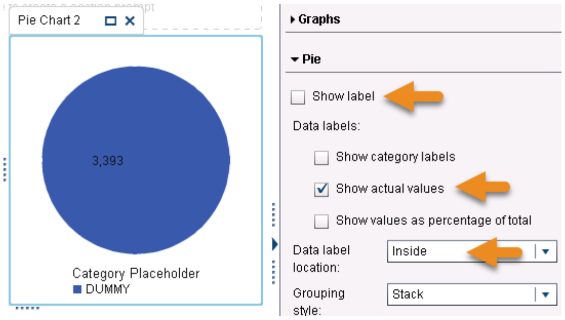


This method retains the value and title but hides the pie itself. Let's build the last one called Unique Customers. For this example, we need a distinct count of the companies that received complaints for the time period and product. This value must be calculated on-the-fly. We can use the Distinct Count feature to count the companies in the dataset. When the viewer selects a value from the drop-downs, the value is immediately re-calculated. Once you have your data item ready to display follow these steps for the pie chart:

1. Create a new pie chart data item. In the Roles tab, add Category Placeholder and Company (Distinct Count) data items.



2. In the Properties tab, do the following:
   a. Remove the check in the Show label checkbox.
   b. Click the Show actual values checkbox.
   c. Change the Data label location to Inside.



☐ Show label

Data labels:

☐ Show category labels

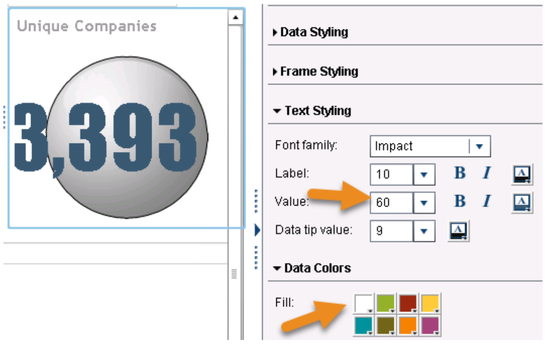
☑ Show actual values

☐ Show values as percentage of total

Data label location: Inside

Grouping style: Stack

3. In the Styles tab, do the following:
   a. In the **Data Styling** area, change the **Data Skin** to **Gloss**. This action causes the HTML5 viewer to display the pie chart as white.
   b. In the Text Styling area, change the Value font size to 60. Adjust the font color and font family if desired.
   c. In the Data Colors area, change the first fill box to White or your background color.
   d. Save your report and look it in the Report Viewer application. With the drop-down filters in place, your new value with change as the selections change.



Font family: Impact

Label: 10

Value: 60

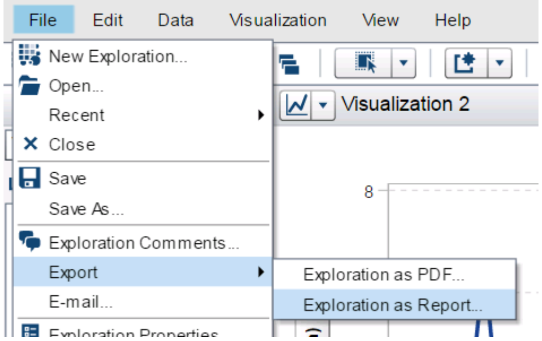
Data tip value: 9

MOVING VISUALIZATIONS TO THE DESIGNER

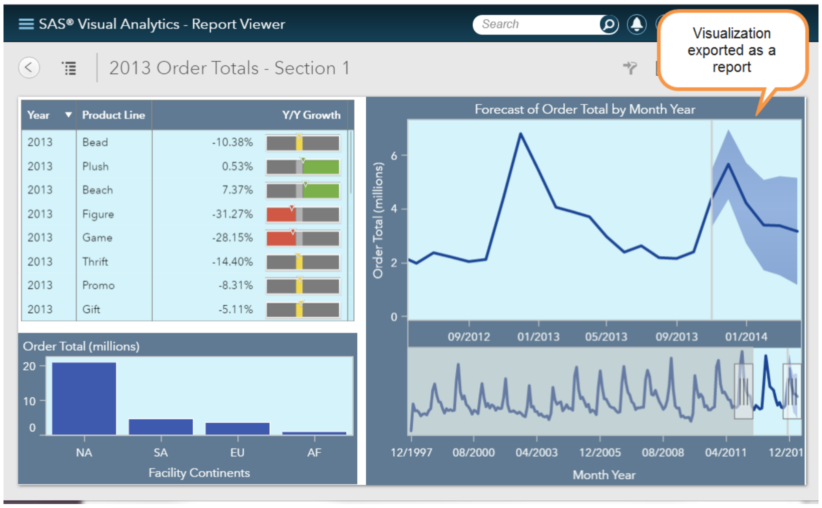
Sometimes you might want to use visualizations from the Explorer in a report so that you can share it with other users. However, if you try to open an exploration in the Designer you won't be able to find it.

When opening in the Designer, the application is looking for report objects. Explorations and reports are saved as different objects in metadata which is why you do not see any of your explorations in the folders. You can get around this problem by exporting a visualization as a report object. Then you can Import it into a report. Here are the steps:

1. Open the Exploration that has the Visualization you want to move into the Designer. Then, go to **File > Export > Export as a Report**
2. Select the visualizations that you want to export as a report. You can select more than one. Click **OK** and then save the Report.
3. From the Designer, open a report. Click the Import tab in the left panel. Choose **Select a report to import** in the drop down and then **Import a report**. Find your visualizations that you saved as a report.



4. Open the report object and you can see your exploration with all your visualizations and their objects listed in a folder structure in the Import tab. Drag the objects into the canvas.



Year	Product Line	Y/Y Growth
2013	Bead	-10.38%
2013	Plush	0.53%
2013	Beach	7.37%
2013	Figure	-31.27%
2013	Game	-28.15%
2013	Thrift	-14.40%
2013	Promo	-8.31%
2013	Gift	-5.11%

Visualization exported as a report

Tip In this figure we have customized the properties and styles to look like the other objects in the report.

In this figure, we brought in a visualization (Line Chart with Forecasting) that was not available in the Designer.

When editing, there is a warning sign in the bottom right corner of the object that lets us know we cannot edit the data. Since these features are not available in the Designer, anything involving the data cannot be changed, so there are no options for the Roles, Display Rules, Filters, etc. You can change how object looks with some options in the Properties and Styles tabs.

WORKING WITH GEOSPATIAL OBJECTS

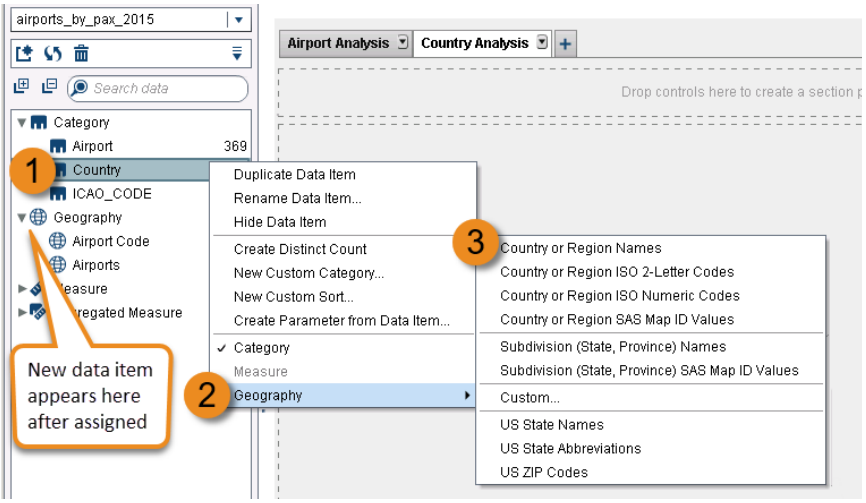
One of the highlights of SAS Visual Analytics is how seamless it is to start data analysis with features like mapping and forecasting. However, with the various formats around the world in location data and advanced scenarios in predicting, using these objects can get overwhelming at times. Here's some information below that could help if you get stuck in those areas.

USING GEO SPATIAL OBJECTS

Before creating a geospatial visualization, you must have a geographic data item. If a data item contains a location, such as a country or state, then it is considered a geographic data item. To keep things easy, SAS Visual Analytics has predefined geographic data elements ranging from the general values, such as country names, to specific values, such as ISO country codes.

To create a geographic data item:

1. Right-click the data item that contains the geographic element that matches the predefined role. In this example, the Country data item contains the country names, such as Australia or Brazil.
2. Select Geography > Country or Region Names. Your data item is moved under the Geography section. You can use it with geographic roles.

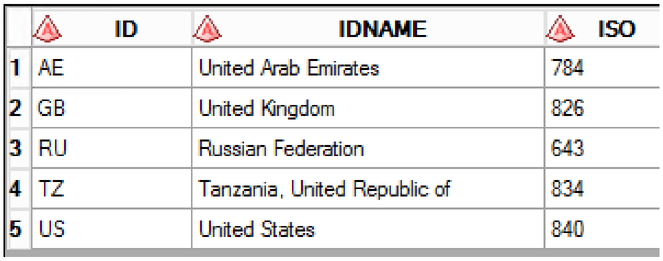


New data item appears here after assigned

Tip On the SAS Support site, there is a Geographic Lookup Values for SAS Visual Analytics webpage that contains a listing of these values to help you understand your specific location. The tables at this site list the countries and the associated ISO numeric codes.

Your Location reference does not match

If your country does not appear in your data object, you might need to convert a country's common name to its official name. For example, Russia, United States of America and Great Britain could be in your dataset, but SAS Visual Analytics is not able to plot them. When you search the country names in the MAPSGFK.WORLD data set, you learn these countries use a different IDNAME.



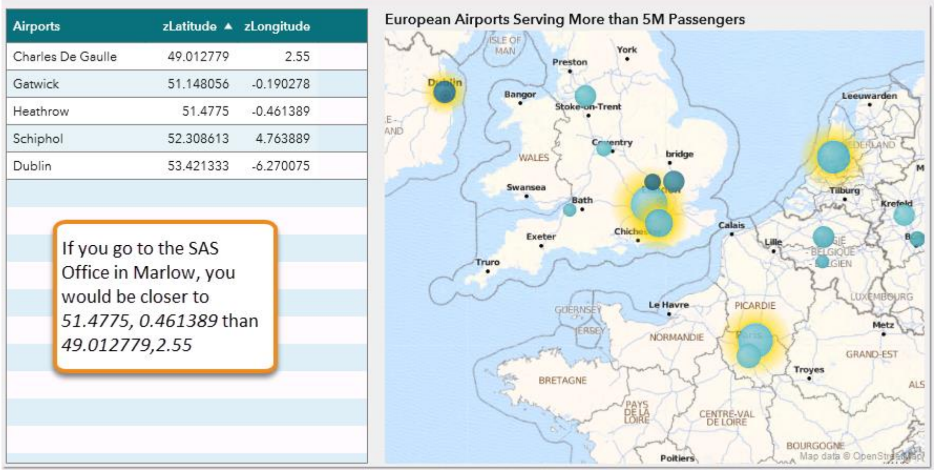
	ID	IDNAME	ISO
1	AE	United Arab Emirates	784
2	GB	United Kingdom	826
3	RU	Russian Federation	643
4	TZ	Tanzania, United Republic of	834
5	US	United States	840

Creating a custom geospatial data item

All geospatial data items represent a location on the Earth. A specific point has a set of coordinates called latitude and longitude. When using a location's latitude and longitude, you are referencing the location.

Locate an Airport

If you think about the world's airports, it's possible to describe the geospatial location with just latitude and longitude coordinates. In this figure, the airports are highlighted on the map and the table on the left shows the airport name with its Latitude and Longitude coordinates.



Airports	zLatitude ▲	zLongitude
Charles De Gaulle	49.012779	2.55
Gatwick	51.148056	-0.190278
Heathrow	51.4775	-0.461389
Schiphol	52.308613	4.763889
Dublin	53.421333	-6.270075

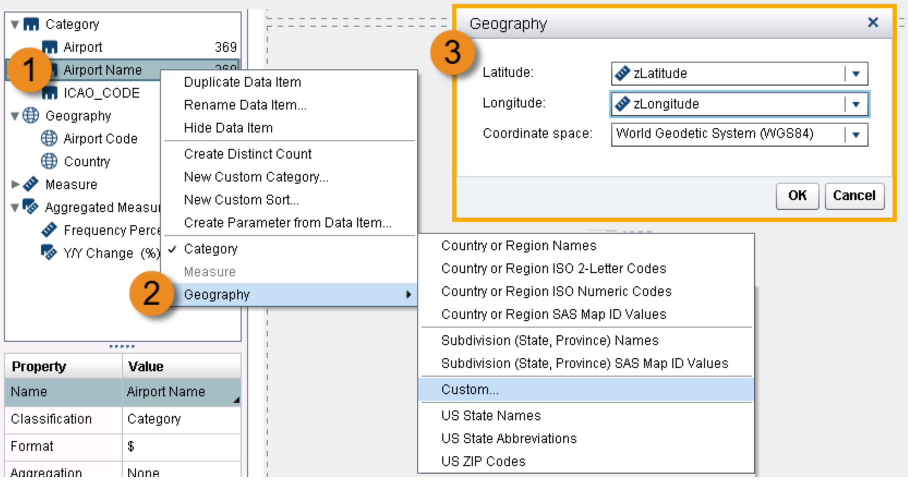
If you go to the SAS Office in Marlow, you would be closer to *51.4775, 0.461389* than *49.012779,2.55*

European Airports Serving More than 5M Passengers

To create a custom geographic item, you must have the latitude and longitude coordinates available in the dataset. Some datasets already have the latitude and longitude available. If you want to add this information you can use open source databases available from a web search.

Here's the steps:

1. Duplicate an existing category data item
2. Right-click the new data item, then select **Geography > Custom**.
3. Select your data items for latitude and longitude in the appropriate fields. Your new data item appears in the Geography area.



Property	Value
Name	Airport Name
Classification	Category
Format	$
Aggregation	None

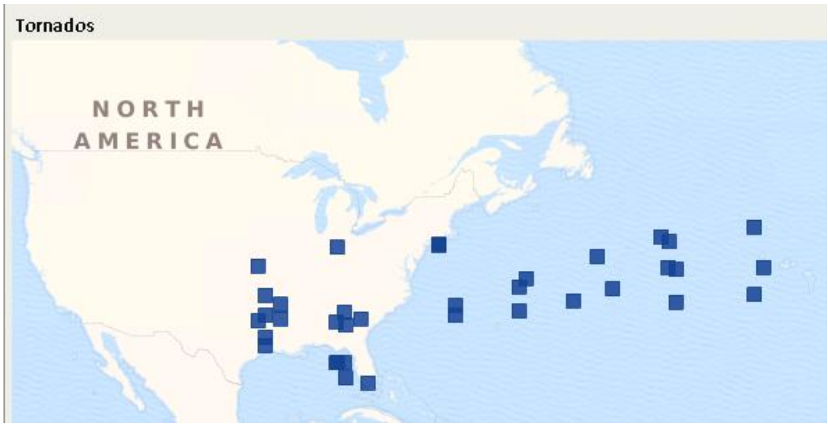
Latitude: zLatitude

Longitude: zLongitude

Coordinate space: World Geodetic System (WGS84)

Tornados in the Atlantic Ocean?

This following figure shows where tornados occurred. It appears they occur in the Atlantic Ocean!



The data table used to create the data object was based on storm events from the past 50 years. However, there must be some typos in the latitude values shifting the storms from the middle of the US to the middle of the ocean.

To correct this situation, the latitude and longitude values would have to be edited. The values would have to be checked against a good source.

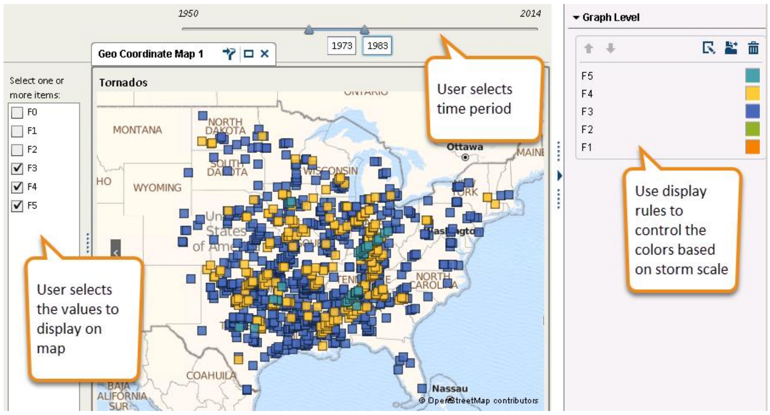
Tip! Use a search engine, such as Google Maps, to check values.

Use filtering to deal with too much data

When you have too much data to display, SAS Visual Analytics issues a yellow icon and warns you to add some filters to your data. With custom geographic data items – it is more likely to happen. The solution is to control how much data appears at once by setting filters.

Here are a few suggested filters:

- Add a date range slider to compare events along a time scale. Adding Event Year to the slider allows the user to compare which years may have had more active storm seasons.
- Split the data item categories. Use the display rules to assign the tornado scale to as a different color so each level is more clear. Then add a List filter and assign the Tornado F/EF Scale to the list. Users can select which tornado scale they want to compare.



User selects time period

User selects the values to display on map

Use display rules to control the colors based on storm scale

USING THE ADVANCED FORECASTING FEATURES

The forecasting feature in SAS Visual Analytics can be applied by users to predict how their data trends into the future. Using data that contains some type of time frame (date or datetime fields), users can use the forecasting option in the explorer line chart object that will model the data to some upcoming time frame. This part is pretty straight forward. However, there are also ways to manipulate the forecasted data to get a better understanding of how certain data items factor into your forecast. Doing so includes finding underlying factors that have a relationship to your forecast which you can then manipulate with scenario analysis.

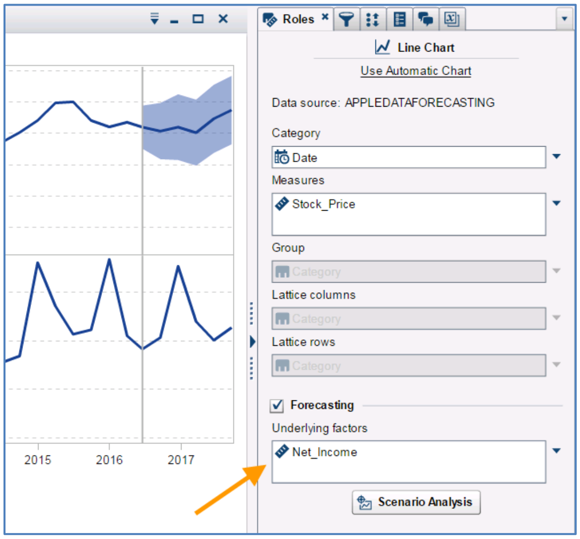
Underlying Factors with Forecasting

In order to improve a forecast, we don't just want to look at one historical measure and base the forecast on those values. There could also be other data points that might have an influence on that measure, and if they are incorporated then our model can become even stronger since it will have multiple variables manipulating it.

The models that SAS Visual Analytics runs in the background to get our forecast can also include other measures into the analysis. This can be done by going to the **Underlying factors** section in the Roles tab. By clicking the drop down, you can add one or more measures from your data set into the analysis.

As with the original forecasting, SAS runs the data through the models to determine the best fit. If the added measure does not have an influence on the model, then it grays out. When the new measure does influence the model, the chart updates with the results as shown in the figure to the right.

Notice how the Scenario Analysis button also becomes available when the underlying factor is run successfully with a data item.



Data source: APPLEDATAFORECASTING

Category: Date

Measures: Stock_Price

Group: Category

Lattice columns: Category

Lattice rows: Category

☑ Forecasting

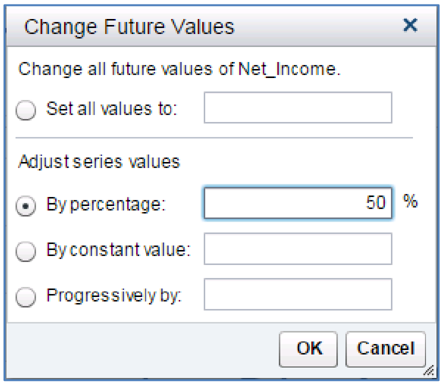
Underlying factors: Net_Income

Using Scenario Analysis and Goal Seeking

Once you have found an underlying factor that influences the forecast, the Scenario Analysis button at the bottom of the Roles tab is available to use. After clicking on it, a window appears that shows the forecasted data field as well as the underlying factor. There are two options here for users to operate and use for analysis, **Goal Seeking** and **Scenario Analysis**.

With Scenario Analysis, users can go in and manipulate the underlying factors and then see how the forecast would change based on those new values. So in our example, we envision that Net_Income is going to increase in the future. We can set this expectation by clicking on the Net_Income button on the left side of the screen and selecting "Set Series Values". A window like the one shown to the right will pop up and this is where the values can be set with a fixed number, a numeric increment, or a percentage increase.

After selecting **Ok**, the forecasted numbers for the Net_Income are updated with the 50% increase.



Change all future values of Net_Income.

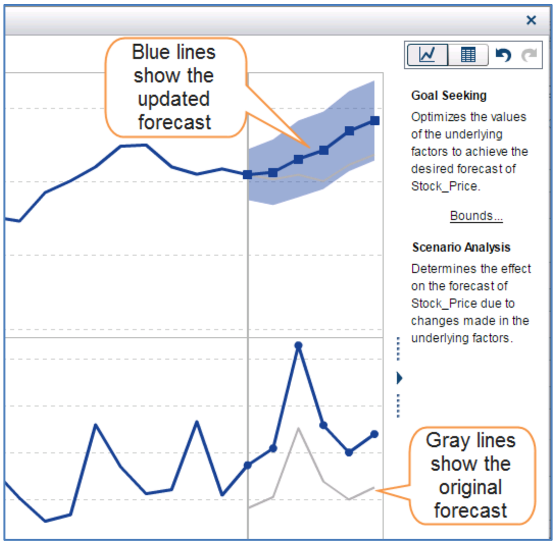
○ Set all values to:

Adjust series values

● By percentage: 50 %

○ By constant value:

○ Progressively by:



Blue lines show the updated forecast

Gray lines show the original forecast

Goal Seeking

Optimizes the values of the underlying factors to achieve the desired forecast of Stock_Price.

Scenario Analysis

Determines the effect on the forecast of Stock_Price due to changes made in the underlying factors.

There is a gray line in the underlying factor's forecast section that indicates the original data points. Since the underlying factor has been altered, only Scenario Analysis is available to use and is the only option available in the right-hand menu.

Goal Seeking works in a similar way except that you are changing the forecasted values and then seeing how the underlying factors would have to change to get those results. Since the underlying factors can have just a small influence on the forecast and they also do not have a confidence range, you will only get an accurate result with something that is heavily correlated.

WORKING WITH THE SAS LASR ANALYTIC SERVER

The SAS LASR Analytic Server is the key component that makes SAS Visual Analytics unique. LASR provides storage and analytic processing of data completely in memory. This lets you analyze and report on data faster than ever. Before the LASR, each time that you queried or performed a calculation in conventional Base SAS, the computer was required to work with chunks of the data by reading it from and writing it to disk repeatedly. While Base SAS is very efficient operating this way, it can still add up to lot of time waiting for large volumes of data since only a relatively small portion of that data can reside in memory at a time. LASR can be deployed as a single service which runs across multiple computers all at once. In this way, LASR can utilize the memory across all of the computers together as a large, scalable work area for storing and processing lots of your data very quickly.

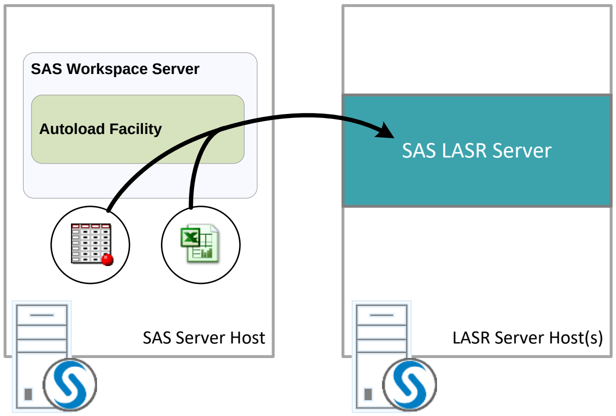
RELOADING DATA

Understanding the SAS LASR Analytic Server requires thinking about data storage in a new way. Long-term storage of data is still on disk, database, or some other data provider. LASR provides a persistent, but not permanent, solution for storing and analyzing data. Because LASR keeps its active tables in computer memory, then when LASR shuts down, those tables effectively disappear. After starting a new LASR, then you must ensure that the data you need is loaded back into memory. There are many ways to load data into LASR, some automatic, some manual, and some programmatic. One of the first techniques you learn when introduced to SAS Visual Analytics is to use the Visual Data Builder interface to import data to the Public LASR Server. Whichever approach you use, data must be reloaded into LASR whenever the LASR Server is stopped and restarted and so SAS provides a two automatic techniques to streamline this responsibility.

SAS Autoload to LASR facility

SAS Visual Analytics offers an Autoload facility for loading data into the SAS LASR Analytic Server. The Autoload facility does not run on LASR's host machines—it is deployed to a host where the SAS Workspace Server for SAS Visual Analytics resides. Each LASR library in your environment will need its own Autoload implementation.

The Autoload facility is scheduled in the host operating system to run at repeated intervals (usually every 15 minutes). It offers the ability to automatically start the associated LASR server (if it is not running already) and then loads, appends, or unloads tables in LASR as directed.



In its current incarnation, the Autoload facility works with data files that reside locally on the SAS Workspace Server host machine in the following formats:

- Base SAS data sets
- Microsoft Excel spreadsheets (XLS, XLSX, XLSB, and XLSM formats)
- CSV (text-delimited file format)

To be clear, the Autoload facility is not suitable for use with data stored in any other medium. It does not work with third-party DBMS tables, SASHDAT tables, nor SAS Scalable Performance Data Engine or SAS Scalable Performance Data Server tables. It also does not integrate in any way with the SAS In-Database Embedded Process.

Keep the following in mind when you're using those capabilities:

- Data must be copied to the Autoload drop zone in a supported format.
- Data is loaded serially to the LASR Analytic Server.

Users of the Autoload Facility must have the ability to create and save files in the physical directories on the host for the SAS Workspace Server. To ensure that users do not clobber each other's files, consider enabling the "sticky bit" on the Autoloader's staging directories.

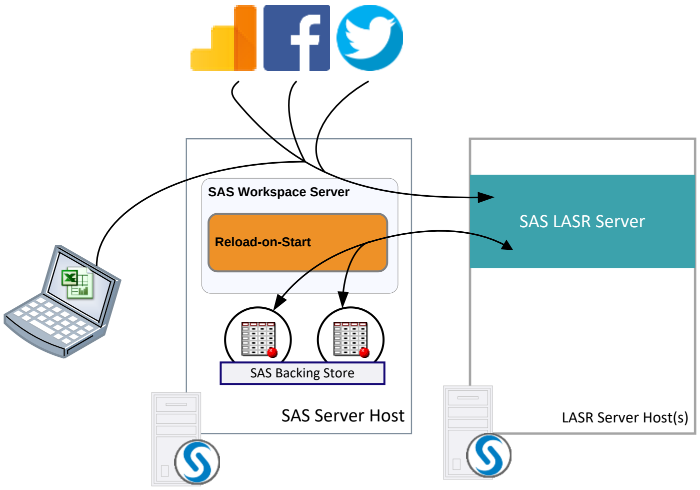
Besides the default provided Autoload for the Public LASR Analytic Server, you have the ability to create additional sources (directories with data on the SAS Workspace Server) and targets (LASR Servers, public or not).

LASR Reload-on-Start feature

SAS Visual Analytics also offers a feature known as Reload-on-Start that automatically reloads previous data back into the SAS LASR Analytic Server when it's started. When enabled, it makes a copy of participating tables and places them in the designated backing store. The backing store keeps copies in only one format: Base SAS data sets. Reload-on-Start is enabled on a per-library basis and supports the option to exclude individual tables as needed.

Reload-on-Start only works with select data sources:

- Imports of local files from users
- Imports of Google Analytics, Facebook, or Twitter data



Like the Autoload facility, Reload-on-Start also does not work with third-party DBMS tables, SASHDAT tables, nor SPD Engine or SPD Server tables. Nor does it integrate with the SAS In-Database Embedded Process.

This feature is intended for relatively smaller size tables. The data that is reloaded to LASR comes only from Base SAS data sets over a single, serial network connection to LASR. As table size increases, watch out for increased disk consumption on the SAS server as well as longer loading times from SAS to LASR.

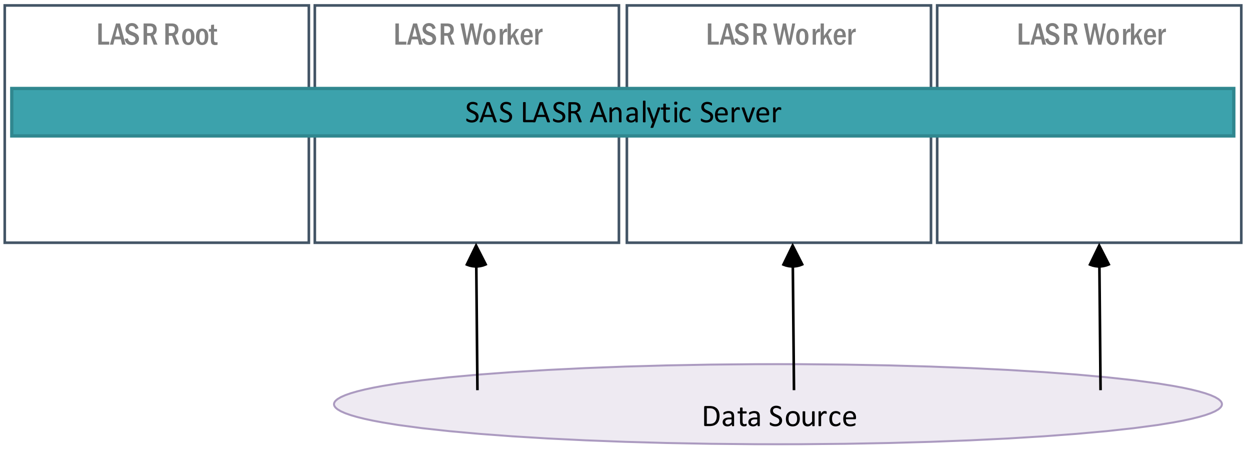
Within these constraints, the Reload-on-Start feature can be very helpful to ensure that the supported tables from the backing store (not from the original upload source) which users expect are indeed available when LASR is (re-)started.

IMPROVING SAS LASR ANALYTIC SERVER PERFORMANCE

The SAS LASR Analytic Server is designed to operate at the speed of electric light. However, with data volumes increasingly getting larger every day, there will be plenty of opportunities for you to improve performance of LASR and realize even more benefits. In particular, there are two main concepts to consider to improve the user experience with LASR performance: loading data and analyzing data.

Improving data loading performance

The conventional approach to loading data into any system is serially. At some point data flows from a source system into the target system through a single point. When the SAS LASR Analytic Server is deployed across many computers at once, then each of the LASR Workers can participate in the data loading process – a technique referred to as a parallel loading.



Right out of the box, LASR can perform parallel loading of plaintext delimited files (like .CSV) as well as the SASHDAT file format from a symmetrically co-located deployment of Hadoop Distributed File System. It can also parallel load from other sources as well, such as SAS Scalable Performance Data Server or third-party database providers when the appropriate SAS In-Database software is available.

Improving data analytics performance

There are a lot of items your information technology organization can consider to optimize your computer environment for best possible performance. For any SAS solution (like SAS Visual Analytics), the SAS Enterprise Excellence Center provides complimentary services to SAS customers to evaluate projected workload and identify the hardware necessary to achieve best performance. Contact your SAS account representative for more information.

Besides in-memory processing, another aspect of SAS LASR Analytic Server performance is that data is distributed across all of the computer hosts running LASR. When an analytic request comes in, each LASR Worker processes its piece of the data and then returns results to the LASR Root for final processing. If there is an increase in the amount of data, or the number of concurrent users, or the complexity of jobs in SAS Visual Analytics beyond your initial estimates, then you should consider adding more computers to your LASR cluster. LASR is designed to scale easily in this way to meet the largest analytic challenges.

To ensure that your day-to-day operations in SAS Visual Analytics continue to perform with expected response times, then consider these ideas to keep things humming:

- Manage your table sizes
  - Don't load any columns or rows you are not using as they will just take up unnecessary space in LASR memory
  - Consider using a custom format for large character based items
  - Very complex custom calculations may impact response time so consider pre-calculating those fields before the data is loaded into LASR
  - Allow plenty of free RAM to exist on your LASR computers beyond storing tables so that LASR has the space it needs to keep track of interim analytic activities

- Manage the SAS LASR Analytic Server
  - Routinely unload any LASR tables from memory that are no longer being used
  - If LASR is symmetrically co-located alongside Hadoop, then consider using SASHDAT files for staging copies of data for LASR
    - SASHDAT provides the fastest, most efficient technique to (re-)load data into LASR
    - LASR can use memory-mapping to SASHDAT directly on disk allowing it to dynamically reallocate RAM to other tasks and still efficiently re-access those tables when needed

CONCLUSION

Getting started with reporting in SAS Visual Analytics is intuitive in many ways. However, once you start to use more complex data or implement different design ideas, then you can run into situations where you might need some help. By scouring through the SAS Communities site we were able to narrow down the topics where users were having the most concern. Those topics, among many others, were then incorporated into this paper and our book, *Introduction to SAS Visual Analytics.* Hopefully, this paper has helped you solve any problems that have come up when using SAS Visual Analytics. If not, you might find the solution on the SAS Communities site or even more in-depth analysis in our book!

REFERENCES

Aanderud, T., and A. Hall. 2012. *The 50 Keys to Learning SAS Stored Processes*. Raleigh, NC: Siamese Publishing.

Aanderud, T., Collum, R., and Kumpfmiller, R., 2017. *Introduction to SAS Visual Analytics*. Raleigh, NC: SAS Press.

SAS Communities - https://communities.sas.com/t5/SAS-Visual-Analytics/bd-p/sas_va

SAS Institute Inc. 2015. *SAS Visual Analytics 7.3: User's Guide*. Cary, NC: SAS Institute Inc.

ACKNOWLEDGMENTS

Thanks to everyone in SAS Visual Analytics Communities for participating. Special thanks to Anna Brown and Chris Hemedinger for providing data and assistance.

We would also like to thank Jaime Thompson for reviewing the paper.

CONTACT INFORMATION

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

Tricia Aanderud
Zencos Consulting
taanderud@zencos.com
http://www.zencos.com

Ryan Kumpfmiller
Zencos Consulting
rkumpfmiller@zencos.com
http://www.zencos.com

Rob Collum
SAS Institute
rob.collum@sas.com
http://www.sas.com