

Five Things You Didn't Know You Could Do with SAS® Visual Analytics

Renato Luppi, Varsha Chawla, SAS Institute Inc., Cary, NC

ABSTRACT

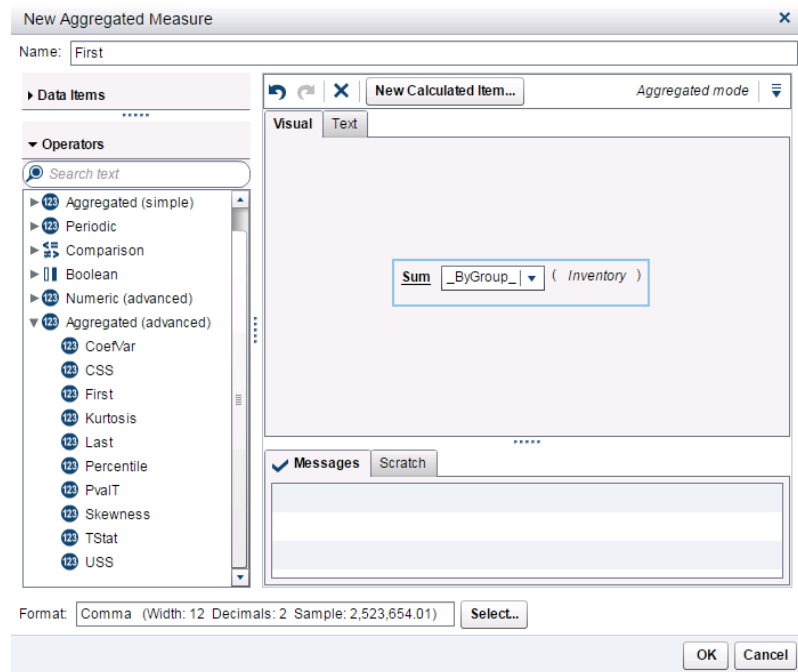
Do you ever wonder how to create a report with weighted averages, or one that displays the last day of the month by default? Do you want to take advantage of the one-click, relative-time calculations available in SAS® Visual Analytics, or learn a few other creative ways to enhance your report? If your answer is yes, then this paper is for you. We not only teach you some new tricks, but the techniques covered here will also help you expand the way you think about SAS® Visual Analytics the next time you are challenged to create a report.

INTRODUCTION

Even if you're a novice SAS® Visual Analytics user, this paper is for you. SAS® Visual Analytics 7.3 contains a great number of built-in features that can simplify the need for worrying about calculations before data is brought into SAS® Visual Analytics. However, many times, users are either unaware of the techniques and operators available to them, or, are not taking advantage of their full potential. We feel that the best way to educate users is to share our experiences in the form of tips and tricks. Many of these tips involve calculations and operators, from one about calculating a weighted average to one that's as simple as right-clicking your mouse. We feel that we have included something for every SAS® Visual Analytics user. We hope you'll agree.

FIRST AND LAST OPERATORS

You're probably familiar with the operators that are available for use in both calculated items and aggregated measures. The simple ones (sum, division, etc.) get used often, but we are going to point out two operators that can really help out with some of the more complex calculations involving measures and date items. The First and Last operators can be found in the expression builder for aggregated measures, under Aggregated (advanced), as shown in Display 1:



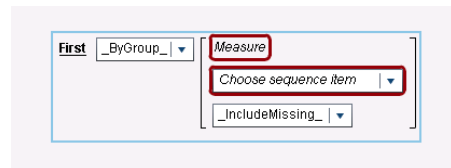
Display 1. The Aggregated Measure Expression Builder

To illustrate how First works, we've created a simple table with dates and inventory amounts:

Date	Inventory
1/17/2017	3215
1/19/2017	2811
3/4/2017	3542
3/3/2017	4111
3/30/2017	4298
4/2/2017	3561
3/11/2017	3000
3/12/2017	2846
3/6/2017	3298
3/15/2017	3124
4/3/2017	3697
3/18/2017	1580
2/23/2017	3255
3/24/2017	3816
2/22/2017	1924
2/25/2017	1687
3/28/2017	1324
4/1/2017	4058

Display 2. Table Containing Dates and Inventory Numbers

Note that these dates and inventory amounts are not in any particular order. When we work with the First operator, we need to have at least one measure in our data. In our case, we have Inventory. We then need a sequence item. Date is perfect for this because it has a natural sequence to it. You could use any date, datetime, or numerical data value here that provides an order to the rows or observations. Finally, we have the option to include or exclude missing items and also to select `_ByGroup_` or `_ForAll_`:



Display 3. First Calculation

Let's work on calculating a few new items to make sense of what we're seeing here. Since we're interested in seeing inventory on a monthly basis, for both the beginning and the end of each month, we first need to make sure that we isolate the months out of the Date variable. This can easily be done by changing the format to: MMYYYY:

Property	Value
Name	Date
Classification	Category
Model type	Continuous
Format	MMYYYY
Aggregation	None

Display 4. Format Change to MMYYYY

Then, we're ready to create the aggregated measures for the inventory on the first and last of each month. Here's how we do this: we select New Aggregated Measure, give it a name, and assign the following variables to it:



Display 5. First Calculation Using Inventory Variable

We do the same for the Last operator:



Display 6. Last Calculation Using Inventory Variable

We change the format to Numeric, which gives us the following crosstab with each month, the inventory on the first date logged of the month, and the inventory on the last date logged of the month.

First and Last Operators		
Date	First Day of Month Inventory	Last Day of Month Inventory
09/2016	2136	1002
10/2016	5483	1010
11/2016	3218	4566
12/2016	2300	5423
01/2017	3215	2811
02/2017	1924	1687
03/2017	4111	4298
04/2017	4058	3697

Display 7. First and Last Results

The First and Last operators are particularly useful for when there is no clear pattern in the sequencing variable/item. Though it would normally be difficult to figure out the first and last days from just the data, these operators make this a breeze. First and Last inventory is just one of the many examples that can be used here. First and Last can also be used to identify and calculate account balances, weather data, patient and attendee volumes, traffic data, and passengers, among many other applications.

PARSE AND FORMAT OPERATORS

Two other operators that we find tremendously useful are Parse and Format. You can use both of these when working with calculated items. According to Google, the word "Parse" is defined as:

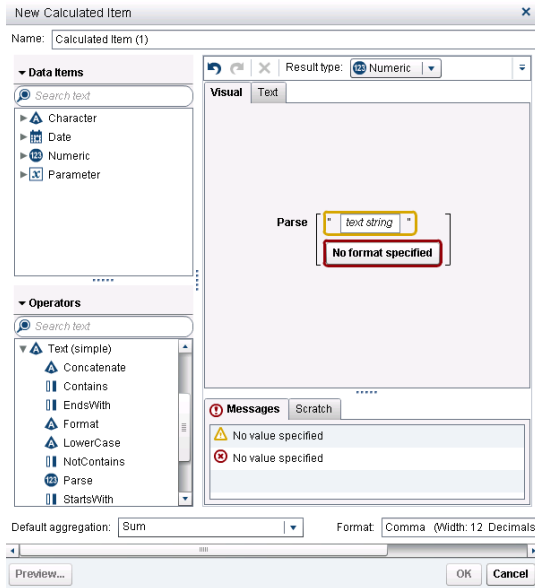
verb

analyze (a sentence) into its parts and describe their syntactic roles

noun

Computing: an act of or the result obtained by parsing a string or a text.

This is exactly what SAS® Visual Analytics does with Parse: it's used to break down a string so that it can be interpreted as the format that you set it to. You can find the Parse operator in the Calculated Item expression builder under Text (simple):



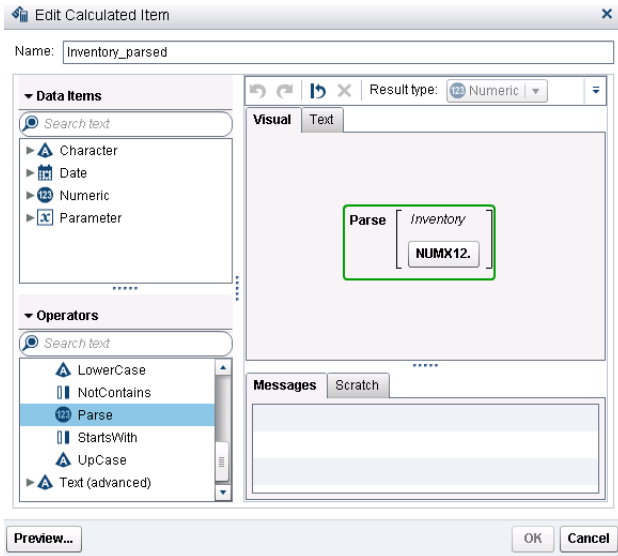
Display 8. Parse Operator

As you can see above, a text string and format have to be specified in order to use Parse. Let's look at a basic example. As you may know, you can change measures to categories, but not the other way around in SAS® Visual Analytics. The Parse operator can do this for us in one simple step. Let's look at the data we're using for this example. Note that this is not the entire data set, but a sample:

Start Date	End Date	Product Category	Sales	Inventory
03Jan2015	12Feb2015	Blue	\$65,000	3547
08Jan2015	12Feb2015	Black	\$72,000	3267
09Jan2015	18Feb2015	White	\$34,000	1258
10Jan2015	19Feb2015	Gray	\$68,000	1574
20Jan2015	26Feb2015	Green	\$48,318	2205
11Feb2015	20Mar2015	Red	\$11,000	6584
14Feb2015	23Mar2015	Blue	\$38,564	1236
19Feb2015	28Mar2015	Black	\$39,882	2544
20Feb2015	29Mar2015	White	\$40,145	4467
21Feb2015	30Mar2015	Gray	\$40,409	2157
01Mar2015	10Apr2015	Yellow	\$50,000	1547
22Mar2015	01Apr2015	Red	\$40,673	9542
25Mar2015	04Apr2015	Blue	\$41,464	9851
30Mar2015	09Apr2015	Black	\$42,782	2546
31Mar2015	10Apr2015	White	\$43,045	9841
01Apr2015	11May2015	Gray	\$43,309	2201
02Apr2015	12May2015	Red	\$43,573	2234
05Apr2015	15May2015	Blue	\$44,364	3301

Display 9. Data Set Sample

The Inventory variable came in as a category instead of a measure. Though the Inventory column looks fine on the surface, we really need it to be numeric so that we can perform calculations on it. We open up a new calculated item expression window and create the following:



Display 10. Parse Calculation with Inventory Variable

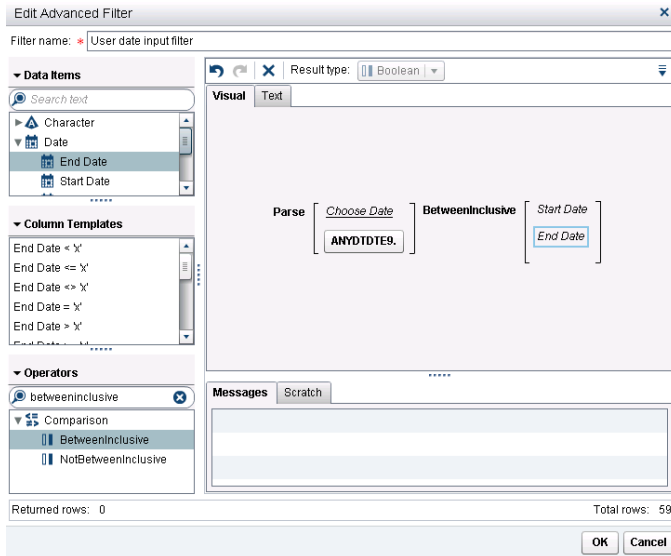
By dragging in the Inventory variable and selecting the desired format, we can now use the newly calculated Inventory_parsed for numeric calculations.

In our second example, we want our report user to be able to pick a historical date to filter the report by. We could take advantage of parameters along with the Parse function to filter the date. For the report user to provide a date, we need both a text input box and a parameter assigned to that text box. Our parameter is a character type without a current value. We added this to the top of our report (which really only contains a list table at this point):

Start Date	End Date	Product Category	Sales	Inventory	Inventory_parsed
03Jan2015	12Feb2015	Blue	\$65,000	3547	3547
08Jan2015	12Feb2015	Black	\$72,000	3267	3267
09Jan2015	18Feb2015	White	\$34,000	1258	1258
10Jan2015	19Feb2015	Gray	\$68,000	1574	1574
20Jan2015	26Feb2015	Green	\$48,318	2205	2205
11Feb2015	20Mar2015	Red	\$11,000	6584	6584
14Feb2015	23Mar2015	Blue	\$38,564	1236	1236
19Feb2015	28Mar2015	Black	\$39,882	2544	2544

Display 11. List Table with Parameterized Text Box

Next we'll create our filter:



Display 12. User Data Input Filter

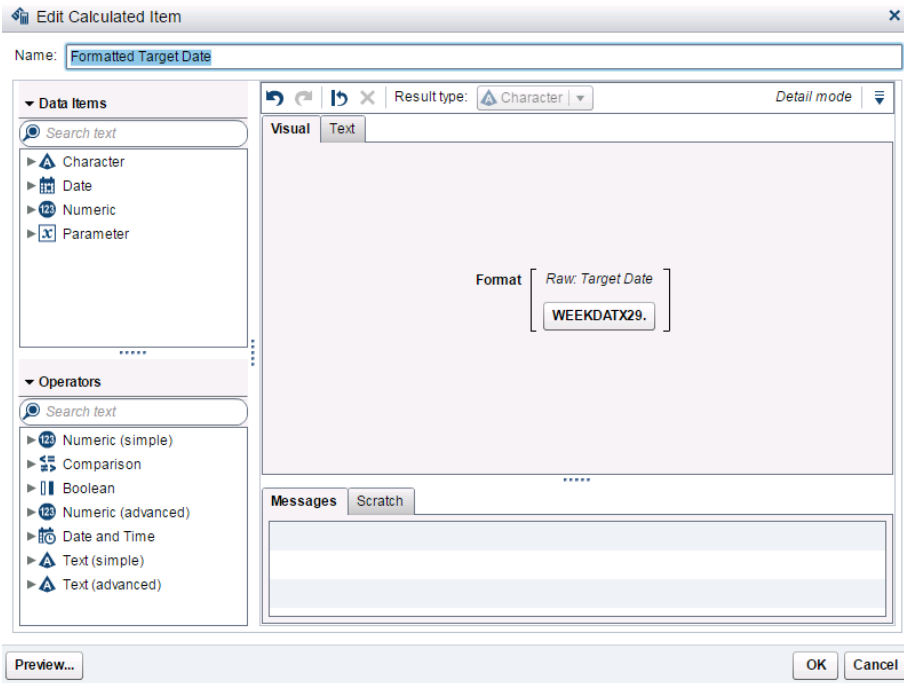
We use the Choose Date parameter as the input string for the Parse operator. We find ANYDTDT9. to be a nice and generic date format to use. The BetweenInclusive operator enables the user to pick any date between the start and end dates. This results in the correct rows being returned when the user inputs a date into the text input box:

Start Date	End Date	Product Category	Sales	Inventory	Inventory_parsed
03Jan2015	12Feb2015	Blue	\$65,000	3547	3547
08Jan2015	12Feb2015	Black	\$72,000	3267	3267
09Jan2015	18Feb2015	White	\$34,000	1258	1258
10Jan2015	19Feb2015	Gray	\$68,000	1574	1574

Display 13. Results Filtered by User Input

The Format function does the opposite of Parse and returns a string instead. This is especially useful when we're working with dates that we need to convert into strings for display purposes. We've used the same data as we used in the Parse example above to convert Target Date from a Date with Month Name format into a WEEKDATX29. format.

We start with a new calculated item and select the Format function. We then drag in the Target Date variable and apply the WEEKDATX29. format, as shown in Display 14:



Display 14. Format Target Date Calculation

This results in a new calculated item that we can then use in a visualization. The big advantage we have here is that we're not just looking at dates, months, and years, but also at the weekdays that match up with these target dates:

Format				
Product Cate...	Inventory	Target Date	Formatted Target Date	Sales
Black	2544	12Mar2015	Thursday, 12 March 2015	\$39,882
	2546	04Apr2015	Saturday, 4 April 2015	\$42,782
	3216	02Jun2016	Thursday, 2 June 2016	\$49,900
	3267	01Feb2015	Sunday, 1 February 2015	\$72,000
	7415	09May2015	Saturday, 9 May 2015	\$47,000
Blue	1236	07Mar2015	Saturday, 7 March 2015	\$38,564
	3126	15Aug2015	Saturday, 15 August 2015	\$48,582
	3210	04Jul2015	Saturday, 4 July 2015	\$45,682
	3301	30Apr2015	Thursday, 30 April 2015	\$44,364
	3547	27Jan2015	Tuesday, 27 January 2015	\$65,000
Gray	9851	29Mar2015	Sunday, 29 March 2015	\$41,464
	1266	30Jun2016	Thursday, 30 June 2016	\$50,427
	1574	03Feb2015	Tuesday, 3 February 2015	\$68,000
	2157	15Mar2015	Sunday, 15 March 2015	\$40,409
	2201	26Apr2015	Sunday, 26 April 2015	\$43,309
	5411	11Jun2015	Thursday, 11 June 2015	\$47,527

Display 15. Final Result

As you can see, Parse and Format come in handy in many cases. We also see these operators used quite a bit when dates need to be used as parameters. You can use Format to populate a dropdown with dates and Parse to take the selected value back to a date used in an expression. An example of this is when the user wants to display the dates before and after a selected date, or wants to put the current

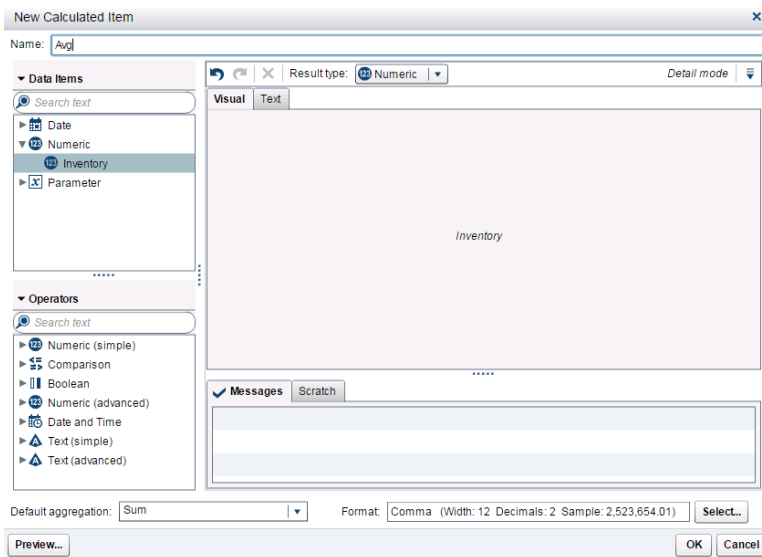
month and the same month last year side by side in a Crosstab or List table. When it comes to dates, these operators can be used any time you're looking to use a month / date that is not exactly the selected month / date. It's a great way to make reports more flexible.

WEIGHTED AVERAGE CALCULATION

SAS® Visual Analytics provides several built-in calculations, including averages and medians. When you click on a measure in either the Explorer or Designer, you can change the aggregation to “Average” by clicking on the “Value” next to the “Aggregation” property and changing it to “Average.” In addition, you can calculate the average by creating an aggregated measure and then selecting the “Avg” operator under “Aggregated (simple).” Both of these examples are shown in Display 16:

Property	Value
Name	Value
Classification	Measure
Model type	Continuous
Format	Numeric
Aggregation	Average

Display 16. Automatic Average Calculation



Display 17. Average Calculation through Aggregated Measure

Now what if you want to calculate a weighted average? The more common calculation of average is the arithmetic mean, which assumes that each data point has an equal weight. As we've seen above, SAS® Visual Analytics can easily calculate this. Unlike the arithmetic mean, the weighted average takes into account the fact that some data points contribute more to the average than others. The calculations make use of relative weights that are assigned to each data point.

A weighted average is often used in Finance. One example is determining the number of shares outstanding in an organization. To get an accurate picture of these shares, the weighted average number of shares can be calculated by taking the number of outstanding shares and multiplying the portion of the reporting period covered. When you do this to each portion and sum the total, you obtain the final

weighted average of shares. The downside of not using weighted averages is that otherwise, only the starting or ending number of shares for the year would be counted.

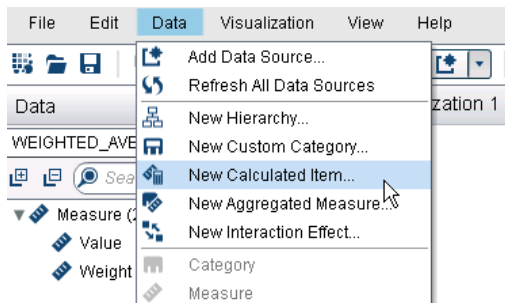
In another example, many universities use a weighted average for calculating grades and GPAs. Many colleges put more weight on core classes than those not required for a major or concentration. A weighted average can reveal a more accurate GPA, focusing on those grades that matter most.

In a few easy steps, the weighted average can also be calculated in SAS® Visual Analytics. Here’s how this works. First, if you’re interested in the weighted average in the first place, your data will likely have a column that represents some kind of weight, and a column that represents a value of something. These two columns are required in order to complete a weighted average calculation. We’ve included our 9 rows of data to show you a very simple example that you can follow along with:

	A	B
1	Value	Weight
2	11	4
3	15	2
4	16	11
5	20	9
6	21	5
7	29	1
8	30	2
9	31	4
10	32	5

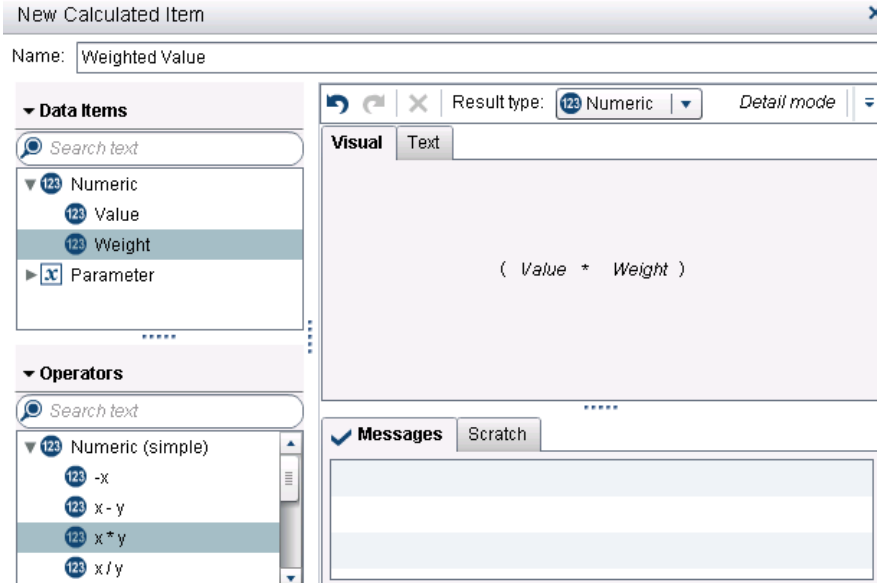
Display 18. Value and Weight Table

1. You can choose to import your data using SAS Visual Data Builder, or load it into SAS LASR Analytic Server through the SAS Visual Analytics Administrator. Once your data has been loaded, open up a new Exploration window and select “New Calculated Item” under the “Data” menu item, as shown in Display 19:



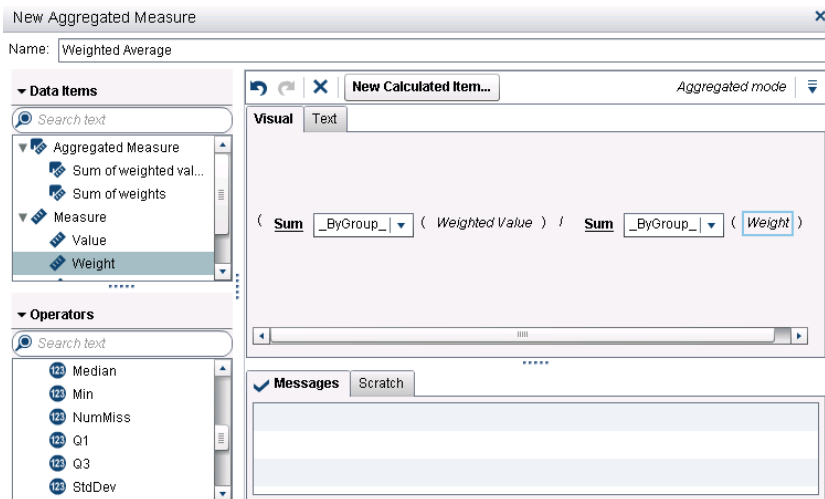
Display 19. New Calculated Item

2. Call your new Calculated Item “Weighted Value.” This new item will be calculated by multiplying your values by your weights, as shown in Display 20:



Display 20. Weighted Value Calculation

3. You are now ready to create the weighted average. Create a new aggregated measure, using the same menu as you used in Step 1. Divide the sum of weighted values by the sum of weights using the `_ByGroup_` options. Rename the measure to “Weighted Average”:

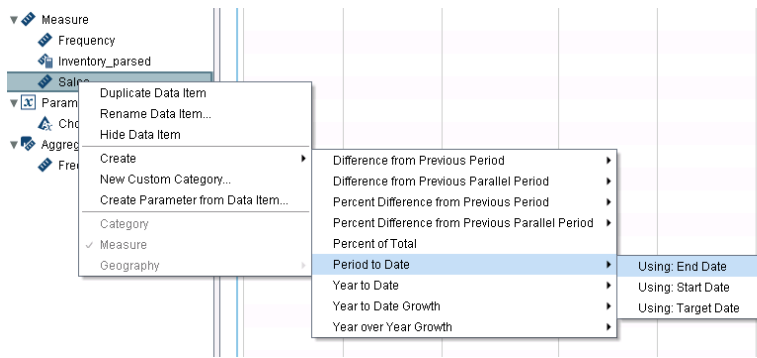


Display 21. Weighted Average Calculation

Want to test your results? Simply drag Weighted Average into a crosstab to check your numbers. If you want to break down this calculation even more, you could create the sum of weighted values and the sum of weights as separate aggregated measures as well.

CLICKING THE RIGHT MOUSE BUTTON

Have you ever right-clicked on a measure in SAS® Visual Analytics? If not, you’re in for a treat. Try right-clicking on any measures, then selecting Create, and you will unveil the following:



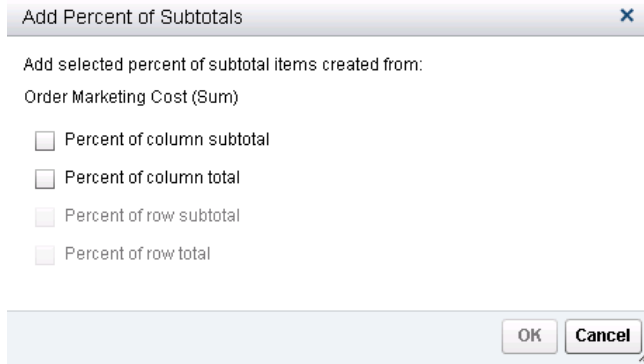
Display 22. Right Mouse Button Options

The aggregated measures shown here are built into SAS® Visual Analytics so that you do not have to create them. Simply right-click and drag them onto a visualization. Many users are aware of this convenience, but there are additional aggregated measures available when you're working in a crosstab. Let's look at a simple crosstab (either in the Explorer or Report Designer, as shown in Display 23) containing continent, product brand, product line, order sales, marketing, and total cost information. When we right-click on the **Order Marketing Total** column and then select Create and Add, we can see that in addition to the options that we saw above, we also see that Percent of Subtotals is listed:

Facility Conti...	Product Brand	Product Line	Order Total	Order Marketing Cost	Order Sales Cost
Africa	Toy	Figure	3,823,659		
		Game	3,903,770		
		Plush	294,900		
Asia	Toy	Figure	1,197,133		
		Game	1,234,818		
		Plush	83,334		
Europe	Novelty	Beach	10,083,093		
		Bead	15,742,129		
		Thrift	1,582,554		
	Toy	Figure	16,691,573		
		Game	17,336,528		
		Plush	2,109,875		
North America	Novelty	Beach	55,251,762		
		Bead	109,688,725		
		Gift	70,474,951	3,079,499.78	610,116.02
		Promo	36,290,437	1,568,112.19	478,088.06
		Thrift	5,715,739	195,699.45	336,297.74

Display 23. Percent of Subtotals Right Mouse Button Option

Note that for the Create and Add option to be displayed, you need to click on the actual column header. When you click on Percent of Subtotals, the following options are available:



Display 24. Percent of Subtotals Options

Note that the Percent of row subtotal and Percent of row total options are grayed out because row totals and subtotals may not be available, depending on how the crosstab is set up. You can choose to have column and row totals displayed in the Properties tab of the Report Designer.

Let's look at the simple crosstab in Display 25. **Percent of Column Total** and **Percent of Total** are the same in this crosstab, though this is not always the case. We'll take a look at when they are different a little later. If we analyze **Order Marketing Cost (Percent of Column Total)** or **Order Marketing Cost (Percent of Total)**, we can see that these columns are the result of dividing the **Order Marketing Cost** by the **Total** amount for this column, listed at the bottom of the crosstab (the 26,327,535.05 in this case). The **Order Marketing Cost (Percent of Column Subtotal)** is the result of dividing the **Order Marketing Cost** by the subtotal for the **Product Brand**. For example, when we divide the **Order Marketing Cost for North America, Novelty, Thrift**, which is 195,699.45, by the **Subtotal for North America, Novelty**, which is 12,180,354.91, we get approximately the 1.61%, which you find in the cell in the fifth column. This holds true for Product Line because Product Brand is a parent level for Product Line. The percentages at the Product Brand subtotal levels are calculated based on the Facility Continent subtotals, and the percentages at the Facility Continent subtotals are calculated based on the Total.

Facility Conti...	Product Brand	Product Line	Order Marketing Cost	Order Marketing Cost (Percent of Column Subtotal)	Order Marketing Cost (Percent of Column Total)	Order Marketing Cost (Percent of Total)
North America	Novelty	Beach	2,458,531.87	20.18%	9.34%	9.34%
		Bead	4,878,511.62	40.05%	18.53%	18.53%
		Gift	3,079,499.78	25.28%	11.70%	11.70%
		Promo	1,568,112.19	12.87%	5.96%	5.96%
		Thrift	195,699.45	1.61%	0.74%	0.74%
	Subtotal: Novelty		12,180,354.91	51.36%	46.26%	46.26%
	Toy	Figure	5,590,769.33	48.47%	21.24%	21.24%
		Game	5,596,424.97	48.52%	21.26%	21.26%
		Plush	347,126.37	3.01%	1.32%	1.32%
	Subtotal: Toy		11,534,320.67	48.64%	43.81%	43.81%
Subtotal: North America			23,714,675.58	90.08%	90.08%	90.08%
South America	Toy	Figure	1,244,000.73	47.61%	4.73%	4.73%
		Game	1,267,380.81	48.51%	4.81%	4.81%
		Plush	101,477.93	3.88%	0.39%	0.39%
	Subtotal: Toy		2,612,859.47	100.00%	9.92%	9.92%
Subtotal: South America			2,612,859.47	9.92%	9.92%	9.92%
Total			26,327,535.05	100.00%	100.00%	100.00%

Display 25. Calculation Results in Simple Crosstab

These calculations are fairly straightforward. However, when working with a category across the top of the crosstab, we can start to see some more differences between **Percent of Column Total** and **Percent of Total**. In the crosstabs below, Display 26 and 27, we're going to take a look at these metrics for North America (Display 26) and totals for all continents (Display 27). **Order Marketing Cost (Percent of Total)** is calculated by dividing the **Order Marketing Cost** for a particular product line and region by the **Total Order Marketing Cost** for all regions, product lines, and product brands. For example, if you take the **Order Marketing Cost for Novelty, Beach**, which is 2,458,531.87, by the **Total Order Marketing Cost**, which is 29,940,822.90 you get approximately 8.21%. Note that we grabbed the 29,940,822.90 from Display 27, which is part of Display 26. If you were to keep scrolling over to the right in Display 26, you would eventually get to Display 27, which shows Totals for all continents.

Order Marketing Cost (Percent of Column Subtotal) is calculated by dividing the **Order Marketing Cost** for a particular product line and region by the subtotal for that product brand. For example, if you take the **Order Marketing Cost for Novelty, Bead**, which is 4,878,511.62, and divide it by the **Subtotal for Novelty**, which is 12,180,354.91, you get approximately 40.05%. The big advantage of using this calculation is that it is always in relation to a parent, which tends to work well with hierarchies.

Order Marketing Cost (Percent of Column Total) is calculated by dividing the **Order Marketing Cost** for a particular product line and region by the **Total** of both product brands and all product lines. For example, if you take the **Order Marketing Cost for Novelty, Gift**, which is 3,079,499.78 and divide it by 23,714,675.58, you get approximately 12.99%.

Facility Continents ▲		North America				Oceania		
Product Brand ▲	Product Line ▲	Order Marketing Cost	Order Marketing Cost (Percent of Total)	Order Marketing Cost (Percent of Column Subtotal)	Order Marketing Cost (Percent of Column Total)	Order Marketing Cost	Order Marketing Cost (Percent of Total)	Order Marketing Cost (Percent of Column Subtotal)
Novelty	Beach	2,458,531.87	8.21%	20.18%	10.37%	19,755.43	0.07%	
	Bead	4,878,511.62	16.29%	40.05%	20.57%	36,281.81	0.12%	
	Gift	3,079,499.78	10.29%	25.28%	12.99%			
	Promo	1,568,112.19	5.24%	12.87%	6.61%			
	Thrift	195,699.45	0.65%	1.61%	0.83%	247.87	0.00%	
Subtotal: Novelty		12,180,354.91	40.68%	51.36%	51.36%	56,285.12	0.19%	
Toy	Figure	5,590,769.33	18.67%	48.47%	23.58%	28,804.37	0.10%	
	Game	5,596,424.97	18.69%	48.52%	23.60%	30,702.27	0.10%	
	Plush	347,126.37	1.16%	3.01%	1.46%	1,530.86	0.01%	
Subtotal: Toy		11,534,320.67	38.52%	48.64%	48.64%	61,037.50	0.20%	
Total		23,714,675.58	79.21%	100.00%	100.00%	117,322.62	0.39%	

Display 26. Calculation Results in Complex Crosstab

Total			
Order Marketing Cost	Order Marketing Cost (Percent of Total)	Order Marketing Cost (Percent of Column Subtotal)	Order Marketing Cost (Percent of Column Total)
2,939,723.43	9.82%	21.81%	9.82%
5,634,569.68	18.82%	41.80%	18.82%
3,079,499.78	10.29%	22.85%	10.29%
1,568,112.19	5.24%	11.63%	5.24%
257,748.65	0.86%	1.91%	0.86%
13,479,653.73	45.02%	45.02%	45.02%
7,921,050.88	26.46%	48.12%	26.46%
7,992,600.63	26.69%	48.55%	26.69%
547,517.66	1.83%	3.33%	1.83%
16,461,169.17	54.98%	54.98%	54.98%
29,940,822.90	100.00%	100.00%	100.00%

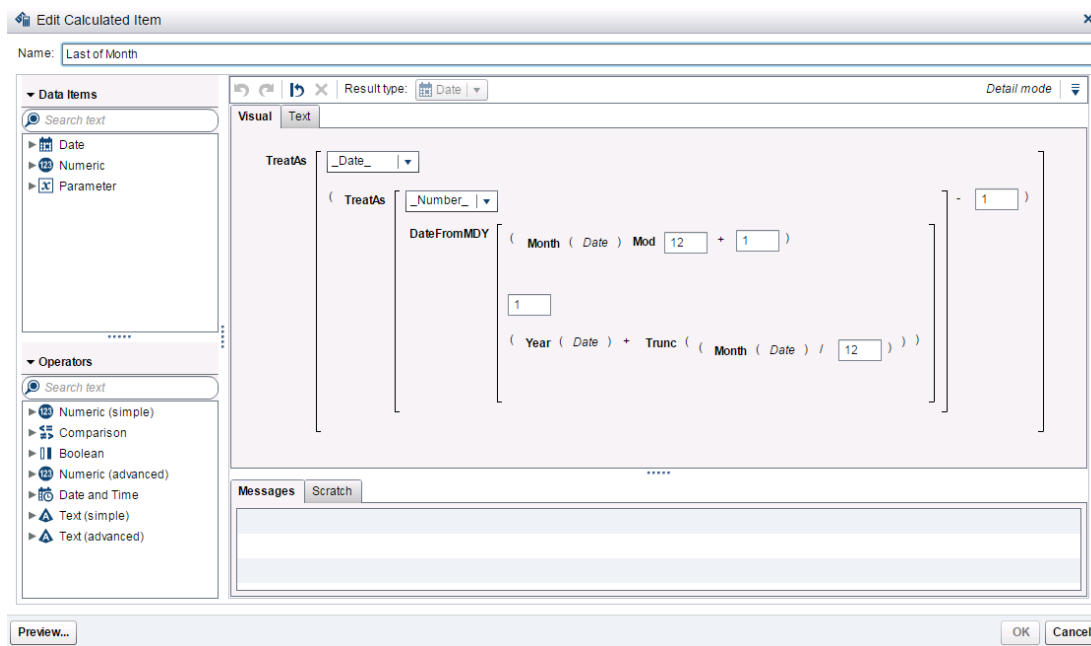
Display 27. Order Marketing Cost Percent Calculation Results

We decided to call out each of these percent calculations because these aggregated measures are visible in the Data tab of the Report Designer, but they cannot be used in any calculations and cannot be edited either. The Percent of Column Total and Percent of Column Subtotal can only be derived from Crosstabs and used in Crosstabs. All calculations cannot be used to aggregate Frequency either, nor can they be used in visualizations besides Crosstabs. Despite these limitations, they are dynamic in the sense that they work for any parent-subgroup data structure.

CALCULATING THE LAST DAY OF THE MONTH

Many times, we have a need for the last day of the month to be included in either visualizations or simple tables. Though it sounds simple, the last day of the month varies according to month and year. Being able to calculate it in a consistent manner can help when figuring out inventory and account balances, processing and due dates, and scheduling tasks.

The last day of the month can be derived in SAS® Visual Analytics by creating a calculated item. The entire calculation is shown in Display 28. We'll break this down into separate components to understand how this is done.



Display 28. Last Day of the Month Example

The concept as a whole is that given a date, you calculate the date equivalent to the first of the next month and subtract one day. This is straight forward for any date in the months from January to November because all you need to do is add one to the month and force the day to be one. However, calculating the first of the next month for a date in the month of December requires some additional logic because December of year Y would become January of year Y+1, after adding one to the month. You could choose to solve this problem using IF-THEN-ELSE logic, but there is a more elegant and efficient way of doing this.

Instead of simply adding one to the month, you add one to (month Mod 12). The operator Mod in this expression returns the remainder of the division of month by 12. This means that for months less than 12, the result of month Mod 12 is the month itself, but for months equal to 12, the result is zero. Adding one to this result will give exactly what you want: the next month.

The sub expression that calculates the year is similar. Trunc (Month(Date) / 12) returns zero for any month less than 12, and returns one if month equals 12. In other words, adding this sub expression to Year(Date) will add one to the year whenever the month is December, which is what you need.

The DateFromMDY() function creates the date corresponding to the first of next month and the TreatAs() functions are used to transform the date into a number, so you can subtract one day and obtain the last of the month, and then to transform that number back into a date.

This may seem like a lot of work for a simple example, but when broken down, is actually quite easy to create and recreate whenever needed.

CONCLUSION

The tips and tricks outlined in this paper are just a few of the many you can find in SAS® Visual Analytics. The more you use the product, the better you become at finding easy ways to enhance your reports. We encourage you to share your findings with others by going to the SAS® Visual Analytics Community at <https://communities.sas.com/>. We hope to find you there soon.

CONTACT INFORMATION

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

Varsha Chawla
SAS Institute Inc.
100 SAS Campus Drive
Cary, NC 27513

Varsha.chawla@sas.com
<http://www.sas.com>

Renato Luppi
SAS Institute Inc.
100 SAS Campus Drive
Cary, NC 27513

Renato.luppi@sas.com
<http://www.sas.com>

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

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