Paper SAS5380-2016

Transform Data Using Expression Builder in SAS® Visual Analytics

Atul Kachare, SAS Institute Inc.

ABSTRACT

Data transformations serve many functions in data analysis, including improving normality of distribution and equalizing variance to meet assumptions and improve effective sizes. Traditionally, the first step in the analysis is to preprocess and transform the data to derive different representations for further exploration. But now in this era of big data, it is not always feasible to have transformed data available beforehand. Analysts need to conduct exploratory data analysis and subsequently transform data on the fly according to their needs.

SAS® Visual Analytics has an expression builder component Integrated into Visual Data Builder, Visual Analytics Explorer, and Visual Analytics Designer that helps you transform data on the fly. The expression builder enables you to create expressions that you can use to aggregate columns, perform multiple operations on data, and perform conditional processing. It supports numeric, comparison, Boolean, text, and date time operators, and different functions like Log, Ln, Mod, Exp, Power, Root, and so on.

This paper demonstrates how you can use the expression builder that is integrated into the data builder, the explorer, and the designer to create different types of expressions and transform data for analysis and reporting purpose.

INTRODUCTION

Data transformation is the process of converting data from one format, scale or distribution to another, in order to generate actionable information to drive smarter decisions and empower business users to diagnose and solve problems. Transformations are usually applied so that data can closely meet the assumptions of statistical inference procedures, or to improve the interoperability or appearance of graphs. When there is evidence of substantial skew in the data, it is common to transform the data to symmetric distribution before constructing a confidence interval so that the resulting confidence interval will likely have a better coverage probability. Data can also be transformed to spread points more uniformly and make it easier to visualize them.

Beginning with SAS® Visual Analytics 6.1, the suite of products has a point and click expression builder interface that enables users to build expressions to transform the data. Thus, they can create new data items from an existing data source without leaving the application; this makes data preparation, exploring, modelling and reporting seamless.

Expression builder has a rich set of operators or functions grouped under different categories to create calculated Items, aggregate columns, perform conditional processing, string manipulations and filter data records.

SAS VISUAL ANALYTICS

SAS® Visual Analytics is an easy-to-use, web-based product that uses SAS high-performance analytic technologies. It empowers organizations to explore huge volumes of data very quickly to identify patterns, trends, and opportunities for further analysis. SAS® Visual Data Builder (the data builder) enables users to summarize data, join data, and enhance the predictive power of their data. Users can prepare data for exploration and mining quickly and easily. The highly visual, drag-and-drop data interface of SAS® Visual Analytics Explorer (the explorer), combined with the speed of the SAS® LASR™ Analytic Server, accelerates analytic computations and enables organizations to derive value from massive amounts of data. This creates an unprecedented ability to solve difficult problems, improve business performance, predict future performance, and mitigate risk rapidly and confidently. SAS® Visual Analytics Designer (the designer) enables users to quickly create reports or dashboards, which can be viewed on a mobile device or on the web.

Starting in the 7.2 release, the explorer enables you to create, test, and compare models based on the patterns discovered during exploration of the data. The explorer enables you to explore, discover, and predict using your data. You can export the score code, before or after performing model comparison, for use with other SAS products and to put the model into production.

SAS Visual Analytics empowers business users, business analysts, and IT administrators to accomplish tasks from an integrated suite of applications that are accessed from a home page. The central entry point for SAS Visual Analytics enables users to perform a wide variety of tasks such as preparing data sources, exploring data and designing reports, as well as analyzing and interpreting data. Most important, reports can be displayed on a mobile device or in the SAS® Visual Analytics Viewer (the viewer).

EXPRESSION BUILDER

SAS® Visual Analytics has an expression builder component integrated into it. It is a drag-and-drop interface that lets you create calculated expressions and add fields to your data source. It is also used for conditional processing and filtering records from the data source. An expression is a calculation used to determine a value and is usually made up of data items and operators or functions. Expression builder is used in different SAS Visual Analytics tools like the data builder, the explorer, and the designer to make it easier for user to transform the data.

DIFFERENT OPTIONS TO LAUNCH EXPRESSION BUILDER

In the designer from the data panel you can launch expression builder to create new calculated items, new aggregated measures or to set new data source filter. Also, if you have already created calculated items or aggregated measures you can right-click on the created item and edit the expression by using expression builder. If you have already set a data source filter then the menu item changes to edit data source filter so that you can edit the filter. The designer also allows you to edit the local data filter set on the object using expression builder. In Figure 1, image on the left shows data panel options to launch expression builder and image on the right shows local data filter edit option to launch expression builder.

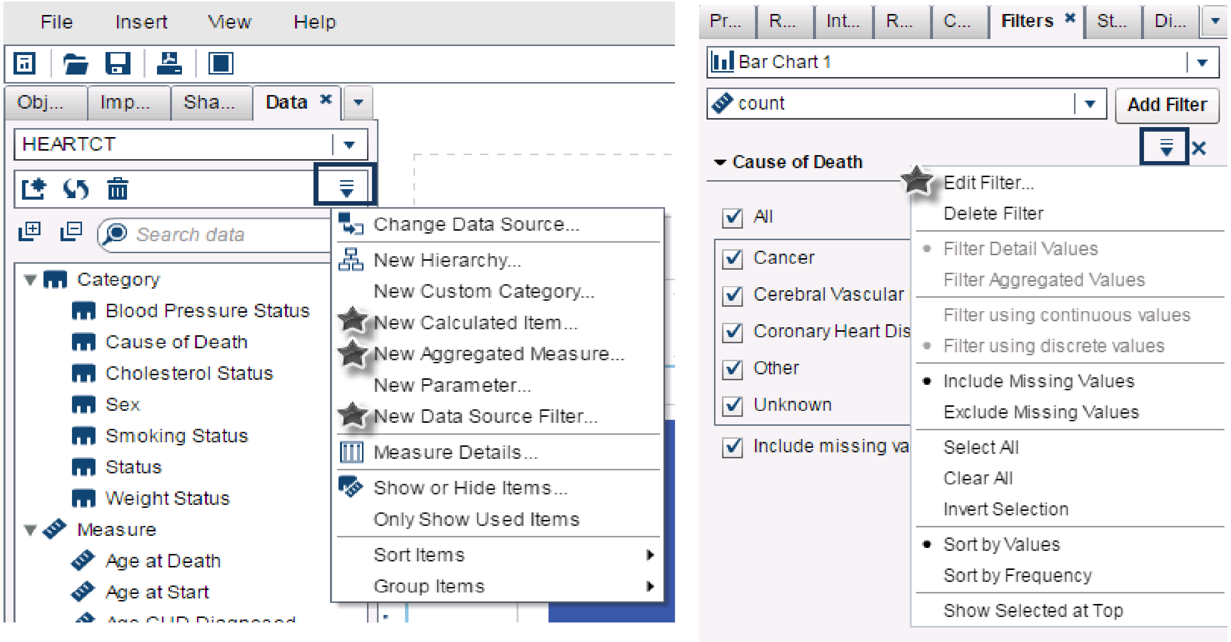


Figure 1 : Launch Expression Builder from the Designer

The explorer also enables you to create calculated items, aggregated measures and data source filters using expression builder. You can also create an advanced filter using expression builder. Figure 2 shows different options available in the explorer to launch expression builder.

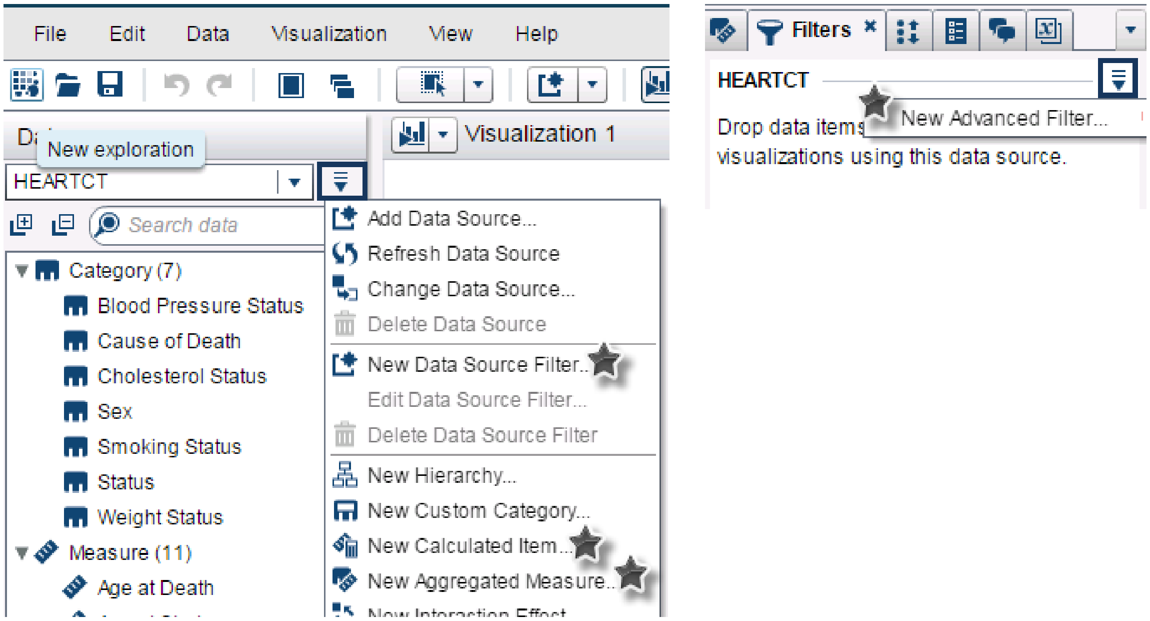
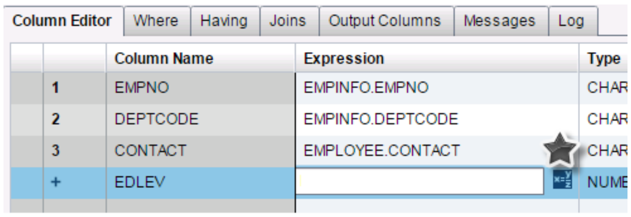


Figure 2: Launch Expression Builder from the Explorer

In the data builder, you can create new columns by specifying an expression to calculate the column value. Click an expression icon in the expression field to access expression builder. There are two ways to filter data in the data builder: WHERE and HAVING clauses. Creating the expression for the filters is similar to creating an expression for a calculated item. Figure 3 shows an option in column editor to launch expression builder and Figure 4 shows the interface you use to build an SQL expression in the data builder.



	Column Name	Expression	Type
1	EMPNO	EMPINFO.EMPNO	CHAR
2	DEPTCODE	EMPINFO.DEPTCODE	CHAR
3	CONTACT	EMPLOYEE.CONTACT	CHAR
+	EDLEV		NUME

Figure 3: Launch Expression Builder from the Data Builder

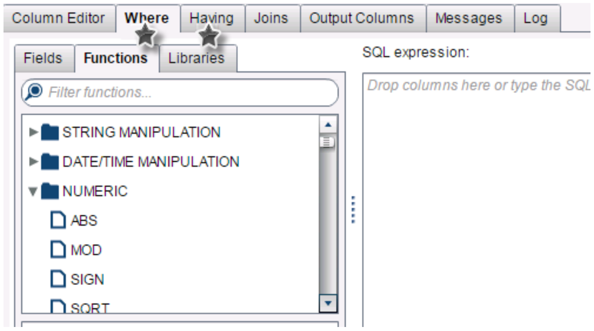
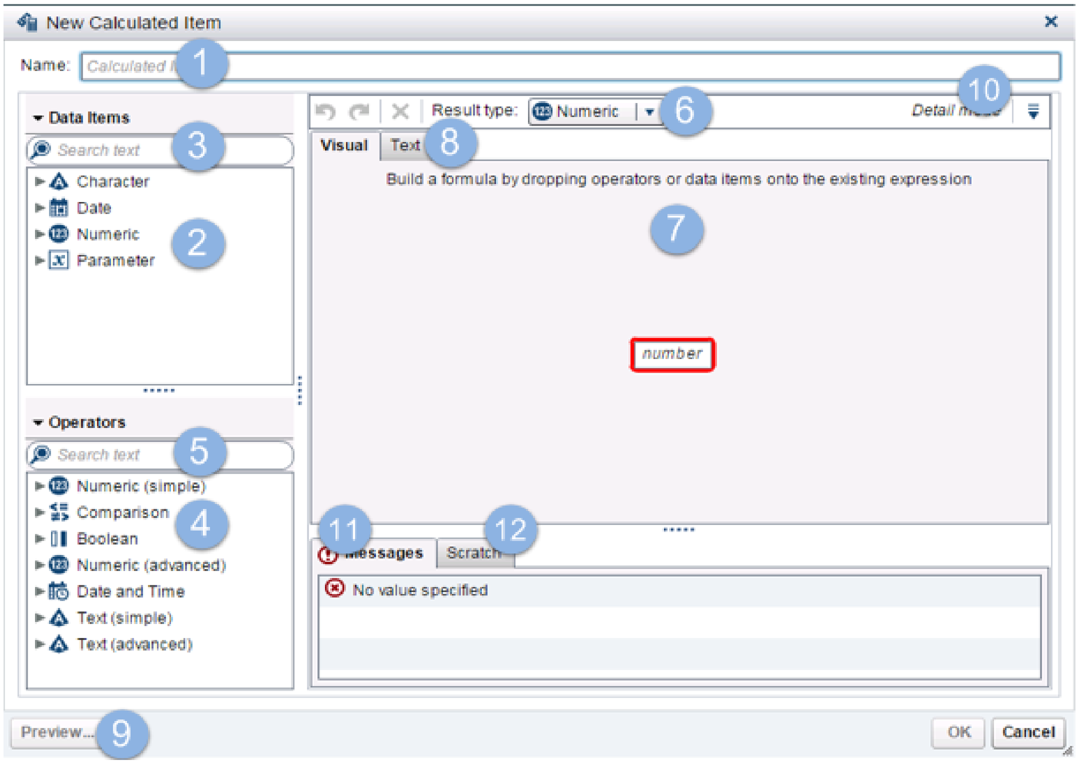


Figure 4: SQL Expression Builder in the Data Builder

DRAG AND DROP OPERATOR AND DATA ITEMS TO CREATE EXPRESSION

The primary way to create expressions in expression builder is to drag operators and data items from the left pane and drop them on the design area. But that is not the only way to create expressions. Advanced users can write expressions directly in the text area. Figure 5 shows the expression builder window for creating a new calculated item.



Build a formula by dropping operators or data items onto the existing expression

No value specified

Figure 5: Expression Builder Window to Create New Calculated Item

1. The name of the expression to be created.
2. The control shows all the data items in the data source available for creating an expression. The data items are grouped in character, numeric and date categories for creating calculated item. For creating aggregated measure they are grouped in aggregated measure, category and measure categories. If you have created parameters then those are listed under parameter category.
3. The search text box to search for variables from the list.
4. The control shows all the operators available for creating calculated Item or aggregated measure. Operators are grouped in different categories and depending on the purpose of launching expression t, different categories are listed
5. The search text box to search for operator from the list.
6. The drop-down list to select the type of the result returned by the expression. This drop-down list is shown for the new calculated item, and return type can be one of the following: character, numeric, date, datetime and time.
7. Build expressions by dropping operators and data items in this area.
8. Using the textual representation of the expression, you can create or modify expressions from here.
9. Preview of the result of the expression.
10. Detail mode has two options to show all drop zones for drag-and-drop operation and showing expression text
11. This area shows error messages related to the expression.
12. Use the scratch area to create or keep an expression before adding it to main expression.

When you are creating a filter expression in addition to above controls, the expression builder lists template expressions for the selected column. Users can directly drag and drop these expressions instead of building new expressions.

CONTEXTUAL OPTIONS TO MODIFY EXPRESSION

You can work with the expression by using the contextual pop-up menu. Right-clicking the operand or operator field in the expression opens a pop-menu that shows options to replace an operand, or operator and add new expression inside as shown in Figure 6. You can use the same menu to copy an expression, move it to the scratch area or remove the expression. You can also preview subexpression results by using this pop-up menu.

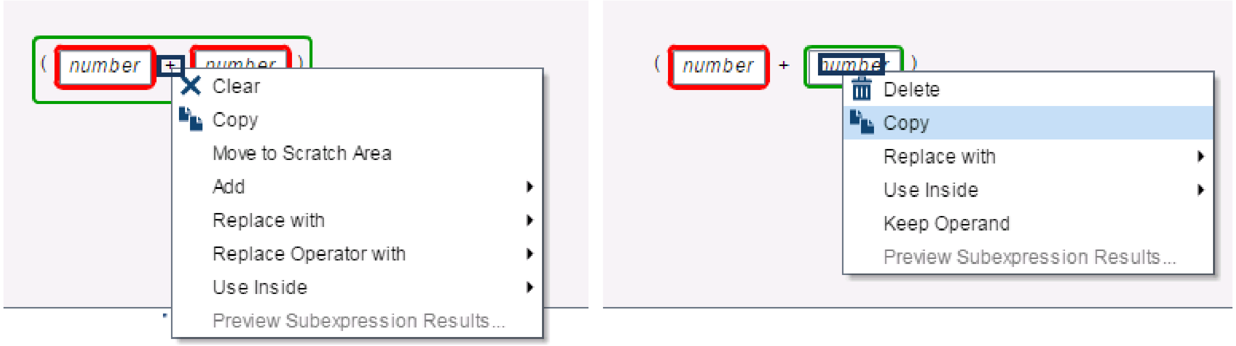


Figure 6: Contextual Options to Work with Expressions

OPERATORS SUPPORTED IN EXPRESSION BUILDER

Expression builder supports numeric, date, Boolean, comparison, conditional, aggregate, periodic, and string operators. This section explains the operators supported in each group.

Numeric Operators

A calculated expression can contain these numeric operators, which take numeric arguments and return numeric results. The argument can be business items, constants, or subexpressions, which themselves return numeric results.

- Simple arithmetic operators -x, x-y, x*y, x/y, x+y.
- Abs – returns the absolute value of the input value.
- Ceil – rounds the input value up to the nearest integer.
- Exp – raises the constant *e* to the power specified by the input value.
- Floor – rounds the input value down to the nearest integer.
- Ln – returns the natural logarithm (base *e*) of input value.
- Log – returns the logarithm of the first value, where the second value specifies the base.
- Mod – returns the remainder after dividing the first value by the second value.
- Power – raises the first value to the power of the second value.
- Root – returns the *n*th root of the first value, where the second value specifies *n* (the base of the root)
- Round – rounds the first value to the number of decimal places that is specified by the second value. Select the second value from the drop-down list.
- TreatAs – allows a numeric, date, or datetime value to be used as different data type within other operators. You can select one of the following: _Date_, _Datetime_, _Number_, Time_.
- Trunc – truncates the input value to an integer.

Date Operators

A calculated expression can contain these date operators, which take numeric arguments and return numeric results. This arguments can be business items, constants, or sub-expressions, which themselves return numeric results.

- DateFromMDY - creates a date value from separate month, day, and year values.
- DateFromYQ - creates a date value from separate year and quarter values.
- DatePart – converts a datetime value to a date value.
- DateTimeFromDateHMS - creates a datetime value from a date value and separate hour, minute, and second values.
- DateTimeFromTimeMDY - creates a datetime value from a time value and separate month, day, and year values.
- DayOfMonth - returns the day of month from a date value as a number from 1-31.
- DayOfWeek - returns the day of the week from a date value as a number from 1-7 (1 is Sunday).
- DayOfYear - returns the day of the year from the date value as a number from 1-366.
- Hour - returns the hour from a time or datetime value as a number from 0-23.
- Minute - returns the minute from a time or datetime value as a number from 0-59.
- Month - returns the month from a date value as a number from 1-12.
- Now - creates a datetime value from the current date and time.
- Quarter - returns the quarter from a date value as a number from 1-4.
- Second - returns the second from the time or datetime value as a number from 0-59.
- TimeFromHMS - creates a time value from separate hour, minute and second values.
- TimePart - converts a datetime value to a time value.
- WeekNumber - returns the week of the year as a number from 0-53.
- Year - returns the year from a date value as a four-digit number.

Periodic Operators

Periodic expressions compute the number of transactions within an interval of time bounded on two sides, typically by a year, quarter, or month.

- CumulativePeriod – the aggregated value for a period of time and all of the previous periods of time within a larger period of time.
- ParallelPeriod – returns the aggregated value for a period of time that is parallel to the current period of time.
- Period – returns the aggregated value for period of time.
- PeriodWithDate – returns the aggregated value for a specific, constant period of time.
- RelativePeriod – returns the aggregated value for a period of time that is relative to current period.

Aggregate Operators

Aggregate operators are used to aggregate results of a query for data items in the data source. Calculated expressions and aggregated expressionscan contain aggregate operators.

- Avg, Count Distinct Max, Median, Min, NumMiss, Q1, Q3, StdDev, StdErr, Sum, Var
- CoefVar – calculates the coefficient of variation of a measure.
- CSS – calculates the corrected sum of squares.
- First – calculates the first value of a measure based on chronological order.
- Kurtosis – calculates the kurtosis of a measure.
- Last – calculates the last value of a measure based on chronological order.
- Percentile – calculates the specified percentile of a measure.
- PvalT – calculates the probability of observing the t statistic value or a more extreme value.
- Skewness – calculates the skewness of a measure.
- TStat – calculates the Student’s *t* statistic for a measure, assuming a mean value of zero.
- USS – calculates the uncorrected sum of squares of a measure.

String Operators

- Concatenate – concatenate two strings.
- Contains - returns true if source string contains the specified string.
- EndsWith - returns true if source string ends with the specified string.
- Format - appliesa format to the source expression and returns a string.
- LowerCase - convertsall alphabetic characters to lowercase.
- NotContains – returns true if source string does not contains specified string.
- Parse – interprets a string according to provided informat.
- StartsWith – determines whether the source string starts with specified string, returning true or false.
- UpCase – converts all alphabetic characters to uppercase.
- FindChar – returns the first index in the source string that matches any of the specified characters.
- FindString – searches string inside another string and returns theindex.
- GetLength – returns the length of the string.
- GetWord – returns the word specified by the index.
- RemoveBlanks – removes leading, trailing, or all blanks from the string.
- RemoveChars – removes specified characters from the string.
- RemoveWord – removes the word specified by the index.
- Replace – searches the string and replaces it with a new string.
- ReplaceWord – replaces a word specified by the index.
- Reverse – reverses the string.
- Substring – extracts characters from a string between two specified indices.
- Update – updates characters from a string between two specified indices by new string.
- URLDecode – decodes a URL encoded string.
- URLEncode – encodes a string using URL encoding to keep it from being confused with the URL itself.

USE OF EXPRESSION BUILDER IN DIFFERENT TOOLS

Example 1: Transform Data for Linearity in the Explorer

Linear least squares regression assumes that the relationship between two variables is linear. In this example, a test is performed on the data to determine if there is a linear relationship between income and infant mortality rate. But a scatter plot with a fit-line for the data shows that there is non-linear relationship between the variables, as shown in Figure 7. Transformations are performed to overcome the non-linearity problem. Taking Logarithm of both income and infant mortality and drawing scatter plot with a fit-line for these two transformed variables shows there is a linear relationship between variables. See Figure 8.

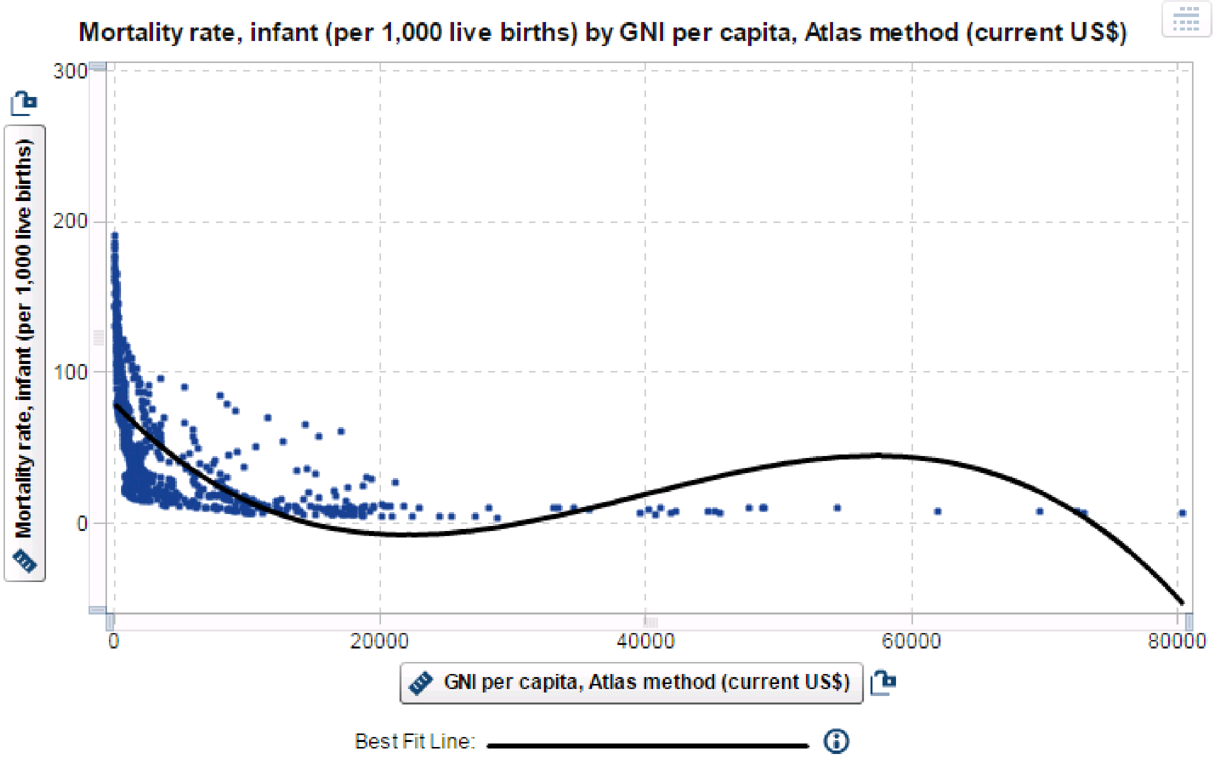


Figure 7: Scatter Plot with a Fit-Line Showing Non-Linear Relationship between GNI Per Capita and Infant Mortality Rate

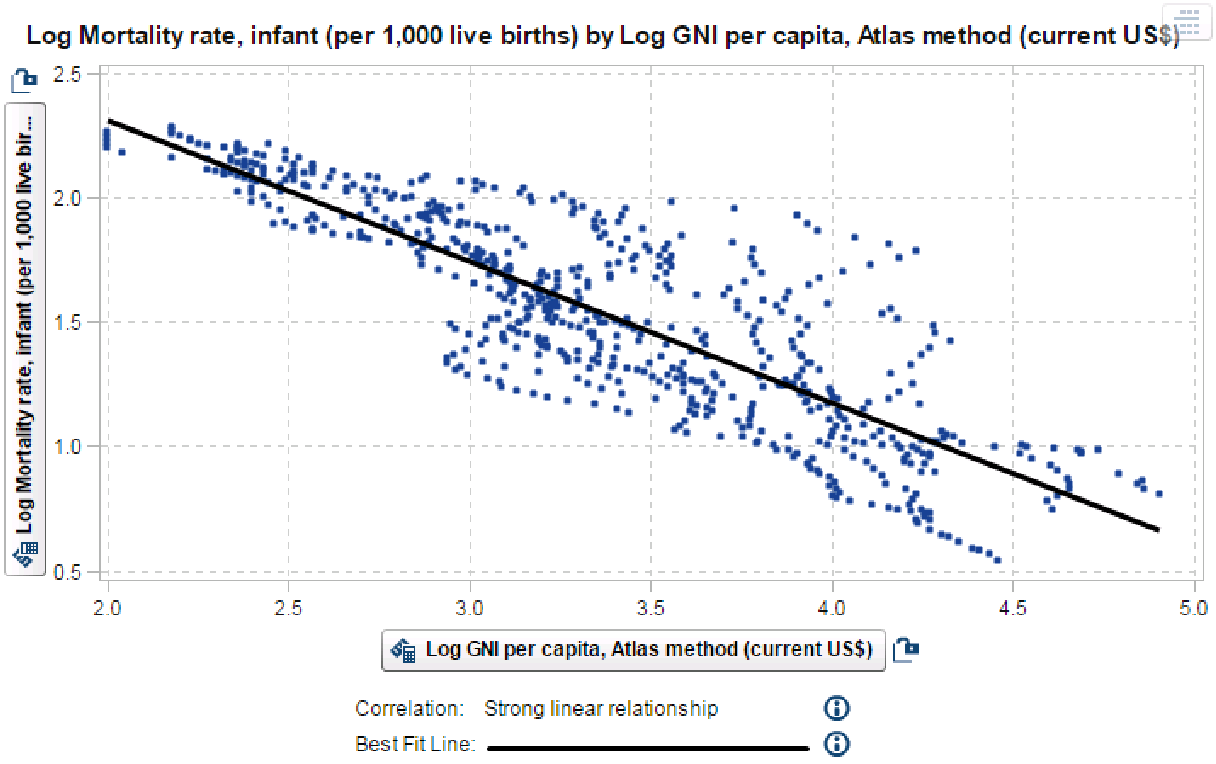


Figure 8: Scatter Plot with a Fit-Line Showing Linear Relationship after Logarithm Transformation

Example 2: Calculate Product Sale Compound Annual Growth Rate using the Designer

The compound annual growth rate (CAGR) is a useful measure of growth over multiple time periods, and it is calculated using a beginning value, ending value, and the time periods. The formula to calculate CAGR is

CAGR = (Ending Value / Beginning Value) $^{(1/\text{ \# of years})}$ - 1

For SAS Visual Analytics, you can calculate yearly CAGR using expression builder. The following example shows the trend in the growth rate of yearly product sales.

Here are the calculated data items:

```
BeginningYearNum = Year('31DEC1998'd)
EndingYearNum = Year('transactionDate'n)
```

Here are the aggregated measure data items:

```
NumYears = Min [_ByGroup_] ('EndingYearNum'n) -
              Min [_ByGroup_]('BeginningYearNum'n)

BeginningValue = PeriodWithDate(_Sum_, 'Product Sale'n, 'transactionDate'n,
                                  _ByYear_, '31DEC1998'd)

EndingValue = Period(_Sum_, 'Product Sale'n, 'transactionDate'n, _ByYear_)

NormalizedRatio = 'EndingValue'n / 'BeginingValue'n

CAGR = ( 'NormalizedRatio'n Power (1 / 'NumYears'n))-1
```

The new aggregated measure CAGR is shown by a dual axis bar-line chart in Figure 9.

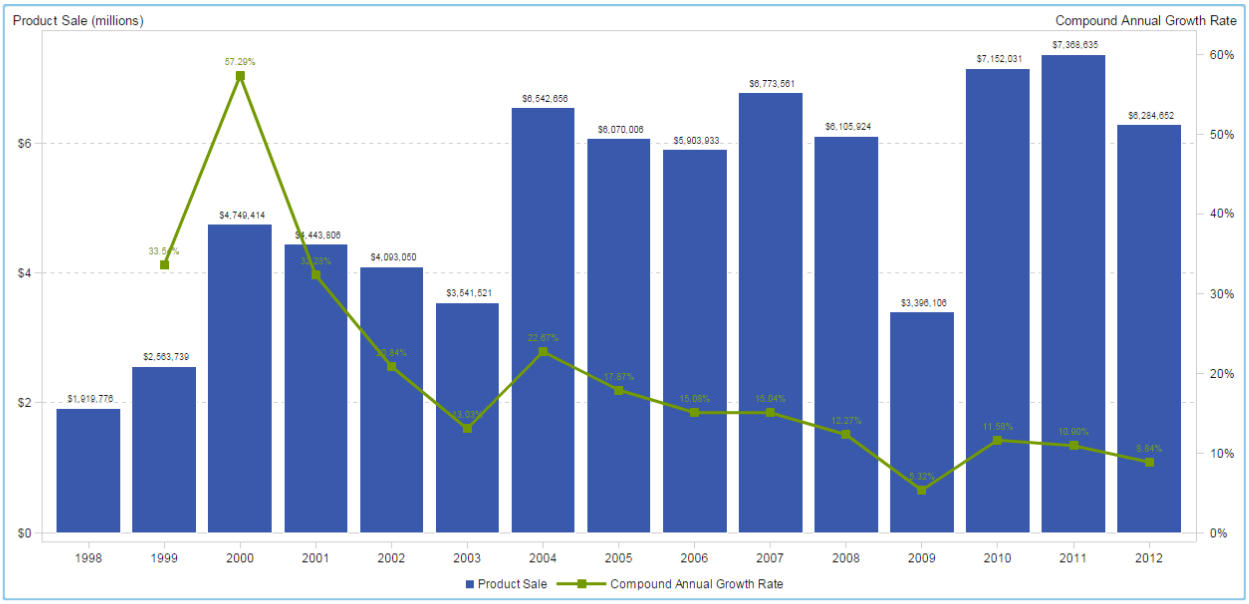


Figure 9: Dual Axis Bar-Line Chart Showing Product Sale and CAGR Per Year.

Example 3: Use of a Conditional Statement to Form a New Calculated Item

This example shows the use of an IF…ELSE conditional statement to create a weight status data item depending on calculated Body Mass Index (BMI).

Here are the calculated items:

```
BMI=( 'Weight'n * 703 ) / ( 'Height'n Power 2 )

WEIGHT_STATUS = IF ( 'BMI'n <= 18.5 )
                   RETURN 'Underweight'
                ELSE (
                   IF ( 'BMI'n BetweenInclusive(18.5, 24.9) )
                      RETURN 'Normal'
                   ELSE (
                      IF ( 'BMI'n BetweenInclusive(25, 29.9) )
                         RETURN 'Overweight'
                      ELSE 'Obese' ) )
```

CONCLUSION

Expression builder's intuitive interface with its rich set of operators enables users to create calculated items, aggregated measures, perform conditional processing and filtering without leaving the application. This makes data preparation, exploring, modelling, and reporting seamless.

RECOMMENDED READING

- SAS Institute Inc. 2015. *SAS Visual Analytics 7.3 User Guide.* Cary, NC: SAS Institute Inc. Available *http://support.sas.com/documentation/cdl/en/vaug/68648/PDF/default/vaug.pdf*
- Chitale, Anand, and Christopher Redpath. 2013. “Whirlwind Tour Around SAS Visual Analytics.” *Proceeding of the SAS Global Forum 2013 Conference.* Cary, NC: SAS Institute Inc. Available http://support.sas.com/resources/papers/proceedings13/057-2013.pdf
- Styll, Rick. 2013. “Fast Dashboards Anywhere with SAS Visual Analytics.” *Proceedings of the SAS Global Forum 2013 Conference*. Cary, NC: SAS Institute Inc. Available *http://support.sas.com/resources/papers/proceedings13/059-2013.pdf*
- *Devarajan, Ravi, et al. 2014.* “Create Custom Graphs in SAS Visual Analytics Using SAS Visual Analytics Graph Builder.” *Proceedings of the SAS Global Forum 2014 Conference.* Cary, NC: SAS Institute Inc. Available http://support.sas.com/resources/papers/proceedings14/SAS346-2014.pdf

CONTACT INFORMATION

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

Atul Kachare
Level 2A & Level 3, Cybercity, Tower 5, Magarpatta city, Hadapsar
Pune, Maharashtra India – 411013
SAS Institute Inc.

atul.kachare@sas.com
http://www.sas.com