Paper 2905-2018

Unleashing the true Power of SAS® For Warranty Analytics for Automobile Industry

Prateek Kumar Singh, Bristlecone India Ltd; Paresh Peter Rodrigues, Bristlecone India Ltd;
Saket Shubham, Bristlecone India Ltd ; Shiv Kumar, Mahindra & Mahindra

ABSTRACT

Performing Warranty Analytics in the Automobile industry can be a cumbersome task as we need a combination of warranty events data along with the production and sales data to understand the rates at which warranty events are received or the payments are made toward those events. However, the data needed to carry out this type of analysis typically comes from different transaction systems. One system records the production and sales of the vehicles and another system contains information about failures and warranty events for these vehicles. By using a combination of SAS products, we integrated the data from these two systems enabling our users to perform the desired warranty analytics. We exploited various unconventional programming techniques to combine and manipulate the data into the desired form for use with SAS® Visual Analytics.

This paper intends to showcase various innovative approaches deployed to treat, manipulate and harmonize loosely related data and effectively facilitate different types of analyses on this data using SAS® Visual Analytics. A distinguished programming methodology has been used in both data manipulation (with Base and Advanced SAS coding) and Report Building (using parameters and calculated items). Developers and consultants can adopt these approaches to create functionalities that aren't directly available in SAS® Visual Analytics 7.3 or earlier versions.

INTRODUCTION

SAS® Visual Analytics (VA) is an integrated suite of web applications that offers intuitive, drag-and-drop interactions, rapid, highly visual responses, and role-based access to functionality. It is used for creating dynamic and interactive reports that can be used for data-backed decision-making, monitoring and control of various operations/processes of an organization.

We have described the data preparation process, creating the visualizations and overcoming the challenges faced in the process. Raw data needs to be treated before it is uploaded on SAS VA for visualizations. While there is a data Building process available in SAS VA where minor manipulations can be made and new variables can be created, it is advisable to prepare the data before it is pushed to VA.

Another way of creating new variables is the "new calculated item" function. It is a handy tool available in the "Report Builder" of SAS VA. Using this function, developers can calculate 'measures' whose values depend upon the selections made by the user (on the fly). What makes this paper and this project unique and interesting is the structure of the data and the analyses enabled using SAS VA.

DATA

INTRODUCTION OF DATA

We extracted a number of tables from the transaction systems. The data in concern can be broadly categorized as:

Events Data

As the name suggests, events data consists of:

1. A set of transaction tables containing warranty events and basic information about them like the complaint group responsible for a particular Event, replaced Part-codes, labour-codes, date of failure etc.
2. Master tables mostly contain descriptions of complaint group-codes, part-codes etc.

Following table represents the events data sans code descriptions:

VIN Number	Claim Id	Group Code	Comp.Code	Part	Event	Cost
ABC	123	G1	C1	P1	1	100
ABC	123	G1	C1	P2		50
ABC	123	G1	C1	P3		50
ABC	123	G1	C2	P4	1	20
ABC	123	G1	C2	P5		30
ABC	123	G2	C3	P6	1	60
ABC	124	G1	C1	P1	1	100

Table 1. Claims data sans code descriptions

Manufactured and Sold Data (MnS)

This is another set of transaction and master tables containing the manufactured and sold details of each vehicle that is produced by the organization and various attributes of the vehicles like production Month and year, plant in which the vehicle was produced, Vehicle Type, Model name, Variant Name, Dealer to which it was billed etc. The total no. of VIN numbers in the data is the number of vehicles sold.

Following table represents the MnS Data:

VIN Number	Prod. Month	Prod. Year	Model	Dealer	Zone
ABC	Apr	2016	YYY	X123	Zone1
PQR	May	2016	QQQ	X212	Zone2

Table 2. MnS Data

Important terms

- Production Month
- Event
- Manufactured and Sold Nos.
- Warranty Cost
- Claim Rate (CR)
- Warranty Cost Per Vehicle (WCPV)

Production Month

Production month is derived from the production date of a vehicle. All the vehicles produced in a particular month are said to be of the same batch. For example, all vehicles produced between 01Jan2016 and 31Jan2016 are said to be of the Batch "Jan16". Production month lies at the base of every analysis in this project. Following is the code to extract Month and year from production date:

```
Prod_month = input (substr (put (prod_date, date9.), 3, 7), monyy7.);
Format Prod_month monyy7.;
```

Event

A Unique combination of Claim Id, Group Code and Comp-Code is considered as 1 Event. An Event can have one or more events depending on the Group Codes and Complaint Codes. In the above table, there are 2 events but 4 events. Further, Multiple parts can be replaced in 1 Event.

Total events for a particular Prod. Month is the no. of events that occurred on vehicles that were produced in that month.

Manufactured and Sold Nos.

Manufactured and Sold Nos. or MnS is the number of vehicles that have been billed to dealers. Total MnS for a particular Prod. Month is the number of vehicles billed out of the vehicles that were produced in that month. For example, let's say that the total no. of vehicles produced in the month of Jan-16 was 2000 and so far 1400 of those vehicles have been billed to dealers. In this case, MnS for Jan-16 is 1400.

Warranty Cost

As the name suggests, warranty cost is the cost incurred toward the events.

It includes the cost of replaced parts as well as the cost of labour.

Note: Our analysis would be based on the no. of events and the warranty cost associated with each Event and not on the no. of events.

Claim Rate (CR)

CR is an estimation of the Event rate. CR for a prod. month is the estimated no. of events per hundred vehicles billed from that month's produce.

CR= (Event/MnS)*100

Warranty Cost Per Vehicle (WCPV)

WCPV is the average warranty cost incurred per vehicles billed from particular prod. month's produce.

WCPV= (Warranty Cost/MnS)

Note: CR and WCPV will be calculated in SAS® Visual Analytics as they are aggregated measures.

REQUIREMENTS

Events data needed to be merged with MnS data to calculate desired KPIs

To calculate CR and WCPV in SAS VA, events data and MnS data needed to be merged together to relate the events to relate the events with the prod. months of the vehicles on which they occurred. Thus the two data sets had to be joined based on the VIN numbers.

Challenge

While a VIN number is unique in the MnS data (as in it is billed only once), in events data, there can be multiple events on the same VIN Number. So when the two data sets are combined, a one-to-many join is executed and the VIN Numbers lose their uniqueness and it becomes a challenge to identify the correct number of vehicles billed.

TIS Buckets

TIS represents “Time in Service” and denotes the no. of months a vehicle has completed in service (post sales to the end customer) before a failure occurs. Every Event on a vehicle is assigned a TIS bucket based on the difference between the date of Event and sale date. Following are the TIS Buckets:

0 TIS – Contains all events that occur before the sale (Pre Delivery Inspections or PDI) of the vehicle or on the same day the vehicle is sold.

1 TIS - Contains all the events that occur within 30 days from sale.

3 TIS - Contains all the events that occur within 90 days from sale.

6 TIS - Contains all the events that occur within 180 days from sale.

12 TIS - Contains all the events that occur within 360 days from sale.

24 TIS - Contains all the events that occur within 720 days from sale.

36 TIS - Contains all the events that occur within 1080 days from sale.

If a vehicle is retailed on 01-Jan-2016 to the end customer and an Event occurs in the vehicle on 15-Feb-2016, then for this Event:

Event date – sale date = 46 days.

Hence, this Event will not appear in of “0 TIS” or “1 TIS” but will be considered as a part of “3 TIS” and all the higher TIS buckets. It can be said that this Event occurred within 12 months of service and within 6 months of service, even within 3 months of service but not within 1 month of service.

TIS	**Months**												
0 TIS	0												
1 TIS	0	1											
3 TIS	0	1	2	3									
6 TIS	0	1	2	3	4	5	6						
12 TIS	0	1	2	3	4	5	6	7	8	9	10	11	12

Table 3. TIS Buckets

The table explains that the events that occurred in the 0th month should be reflected in the 0 TIS, 1 TIS, 3 TIS and so on. Similarly, events that occur in the 1st month should show in 1 TIS, 3 TIS, 6 TIS and so on.

Challenge

a) To achieve the mentioned TIS Buckets, the data needs to be multiplied. Which means that 1 Claim Id will be repeated multiple times in the data. As it is, one Claim Id is repeated as it contains more than one Event and multiple parts. So assuming that the events in Table1 (events data) are 0 TIS events, after adding TIS buckets, the data will look like:(Only adding 0, 1 and 3 TIS for representation)

VIN Number	Claim Id	Group Code	Comp. Code	Part	Event	Cost	TIS Bucket
ABC	123	G1	C1	P1	1	100	0 TIS
ABC	123	G1	C1	P2		50	0 TIS
ABC	123	G1	C1	P3		50	0 TIS
ABC	123	G1	C2	P4	1	20	0 TIS
ABC	123	G1	C2	P5		30	0 TIS
ABC	123	G2	C3	P6	1	60	0 TIS
ABC	124	G1	C1	P1	1	100	0 TIS
ABC	123	G1	C1	P1	1	100	1 TIS
ABC	123	G1	C1	P2		50	1 TIS
ABC	123	G1	C1	P3		50	1 TIS
ABC	123	G1	C2	P4	1	20	1 TIS
ABC	123	G1	C2	P5		30	1 TIS
ABC	123	G2	C3	P6	1	60	1 TIS
ABC	124	G1	C1	P1	1	100	1 TIS
ABC	123	G1	C1	P1	1	100	3 TIS
ABC	123	G1	C1	P2		50	3 TIS
ABC	123	G1	C1	P3		50	3 TIS
ABC	123	G1	C2	P4	1	20	3 TIS
ABC	123	G1	C2	P5		30	3 TIS
ABC	123	G2	C3	P6	1	60	3 TIS
ABC	124	G1	C1	P1	1	100	3 TIS

Table 4. Events data after adding 0, 1 and 3 TIS buckets

b) TIS is an attribute of the events and should not affect the manufactured and sold nos. Meaning, regardless of the TIS selected in the dashboard, the MnS Nos. should remain constant.

SOLUTION FOR DATA-LEVEL CHALLENGES

Solution to the above mentioned challenges lies in the way the data sets (events data and MnS data) are combined and use of flags to identify unique VIN numbers and events.

The following expression illustrates how the unique VIN numbers and events are identified:

Step1

Create a variable to denote the no. of days between the sale date and Event date in the events data with following code:

```
TIS_days = intck (days, sale_date, Event_date);
```

Step2

Create a small data set with the required TIS Buckets and join this data set with the events data to multiply the data into temporary TIS Buckets with following logic:

```
data TIS_buckets;
length TIS_Bucket_ $ 6;
input min max TIS_Bucket_$ ;
infile datalines dlm= ",";
datalines;
0,0,0 TIS
0,30,1 TIS
0,90,3 TIS
0,180,6 TIS
0,360,12 TIS
0,720,24 TIS
0,1080,36 TIS
;
run;
proc sql;
create table events_data1 as
select a.*,b.TIS_bucket_
from events_data as a inner join TIS_buckets as b
on a.TIS_days >= b.min and a.TIS_days <= b.max;
quit;
```

Output 1. Output from a CREATE TABLE Statement

Step3

Combining events data with MnS data: Using these temporary TIS Buckets, create separate data sets for each TIS and combine them with the whole MnS data:

```
%macro TIS(a);

proc sql;

create table join_&a as

select a.*,b.*,"&a TIS" as TIS_Bucket length = 6 from

EVENTS_DATA1 as a right join MnS_data as b on

compress(upcase(a.vin_no)) = compress(upcase(b. vin_Number)) and a.TIS_bucket_ = "&a TIS";

%mend TIS;

%TIS(0);
```

Output 2. Output from a CREATE TABLE Statement in a macro

Step4

Append these data sets together:

```
proc append base=join_0 data=join_1;
proc append base=join_0 data=join_3;
proc append base=join_0 data=join_6;
proc append base=join_0 data=join_12;
run;
```

Output 3. Output from a Append Statement

Step5

Create unique keys to identify events and unique VIN numbers and use them to create flags. (We need to sort the data twice, once by Event key to create Event flag and next by MnS key to create MnS flag. These flags will be used in SAS® VA to calculate Events and MnS Nos.)

```
data PNI_keys;
length MnS_key$ 25 event_key$ 120;
set join_0;
event_key = compress(Group_code||Complaint_Code||TIS_bucket);
MnS_key = compress(VIN||TIS_bucket);
run;

proc sort data=PNI_keys out=PN1;
by MnS_key;
run;

data pni_keys;
set PN1;
by MnS_key;
if first.MnS_key then MnS_flag = 0;
MnS_flag +1;
run;

proc sort data=pnI_keys out=PN2;
by event_key descending Event;
run;

data Data_for_VA;
set PN2;
by event_key;
if first.event_key then event_flag = 0;
event_flag + 1;
run;
```

Output 4. Output from a Data step for creating unique keys

Once the data has been manipulated and formatted as needed, it is uploaded in LASR server to create the dashboard.

REPORT BUILDING

Now that the data is available in SAS® Visual Analytics, we start creating the report for the Business User. However, we face the following challenges while calculating the KPIs and interacting the Control Objects with charts/Graphs:

CHALLENGE WITH EVENT ATTRIBUTE FILTERS

Earlier we saw how the Events Data and MnS data has been combined together into 1 flat table based on VIN numbers. Filtering the data as per the attributes of the events should not have any effect on the number of vehicles billed:

VIN Number	Claim Id	Group Code	Comp. Code	Part	Event	Cost	Cost of Event	TIS Bucket	Prod. Month
ABC	123	G1	C1	P1	1	100	200	3 TIS	Jan - 16
ABC	123	G1	C1	P2		50		3 TIS	Jan - 16
ABC	123	G1	C1	P3		50		3 TIS	Jan - 16
ABC	123	G1	C2	P4	1	20	50	3 TIS	Jan - 16
ABC	123	G1	C2	P5		30		3 TIS	Jan - 16
ABC	123	G2	C3	P6	1	60	60	3 TIS	Jan - 16
ABC	124	G1	C1	P1	1	100	100	3 TIS	Jan - 16
EFG	165	G3	C8	P12	1	120	120	3 TIS	Jan - 16

Table 5. Event attribute filters with no effect on data

Let's assume that we are analyzing the data for Prod. Batch of Jan-16 (Refer data in the above table) and looking at events from 3 TIS. We want to know the Event Rate for this batch.

[CR = (Event/MnS) * 100]

In Table5, we can see that there are 2 unique VIN Numbers denoting that 2 vehicles have been billed from the production month (batch) of Jan-16 and the total no. of events on those vehicles is 5.

So, CR = 5/2 *100 = 250

Now, if we want to analyze the complaint Group G1 (assume that we select G1 from the Group Code Filter), the system would return the following rows:

VIN Number	Claim Id	Group Code	Comp. Code	Part	Event	Cost	TIS Bucket	Prod. Month
ABC	123	G1	C1	P1	1	100	3 TIS	Jan-2016
ABC	123	G1	C1	P2		50	3 TIS	Jan-2016
ABC	123	G1	C1	P3		50	3 TIS	Jan-2016
ABC	123	G1	C2	P4	1	20	3 TIS	Jan-2016
ABC	123	G1	C2	P5		30	3 TIS	Jan-2016
ABC	123	G2	C3	P6	1	60	3 TIS	Jan-2016
ABC	124	G1	C1	P1	1	100	3 TIS	Jan-2016
EFG	165	G3	C8	P12	1	120	3 TIS	Jan-2016

Table 6. Group Code filter for Complaint Group G1

Note that only that VIN number on which an Event was raised under complaint group "G1" is returned now. So the Billed qty. returned by the system now is 1 instead of 2.

So, CR = 3/1 *100 = 300.

Whereas the correct calculation should be:

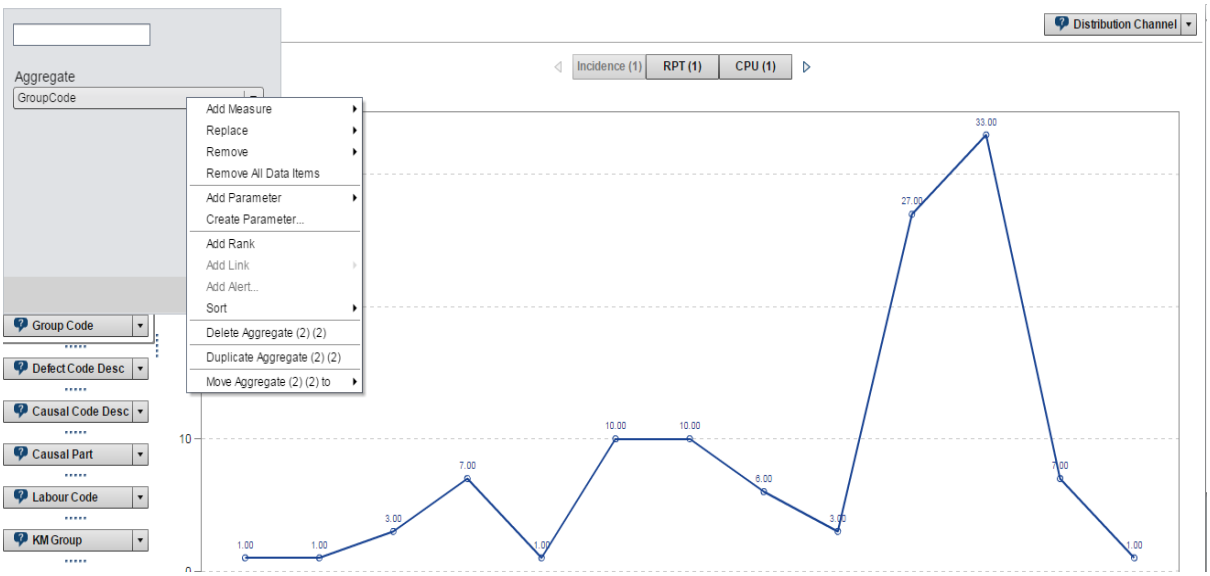
CR = 3/2 * 100 = 150

SOLUTION

To solve this problem, we need to make sure that the control objects containing Event attributes do not filter the chart/graph directly so that MnS can be left unaffected. So, instead of interacting these filters directly with the graphs, we use “Parameters” and “Calculated Items” and calculate the Event in a way that it checks each Event attribute filter to determine if any value has been selected and filters the Event accordingly while having no effect on the MnS Nos.

Step1

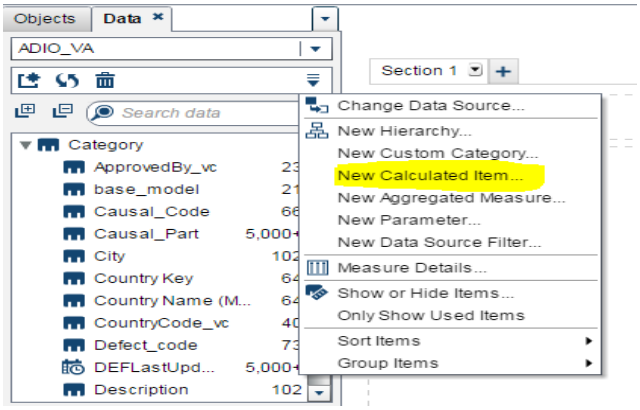
Create Parameters on all the control Objects containing Event attributes.



Display 1. Creating parameter on Claims Filters.

Step2

Create a new calculated item with help of these parameters. Use Nested “IF_RETURN_ELSE” statement as shown in the text box below and use this calculated item as Event in the charts/graphs. This formula may seem too complicated. However, it is simpler to create the IF-RETURN-ELSE nests in the visual mode of calculations in SAS® Visual Analytics.

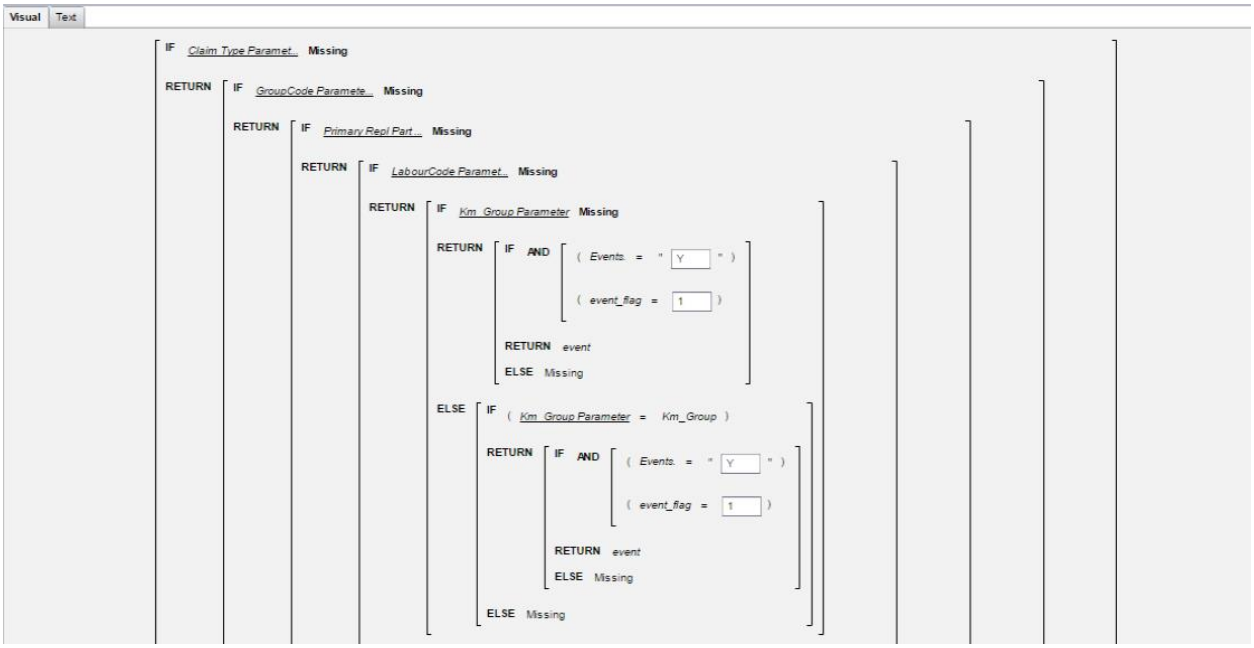


Display 2. Creating new calculated item.

```
IF ( 'GroupCode Parameter'p Missing )
RETURN (   IF ( 'DefectCode Parameter'p Missing )
  RETURN (     IF ( ‘Prim Repl Part Parameter'p Missing )
    RETURN (       IF ( 'event_flag'n = 1 )
      RETURN 'event'n      ELSE . )
    ELSE (       IF ( ' Prim Repl Part Parameter'p = ' Prim Repl Part'n )
      RETURN (         IF ( 'event_flag'n = 1 )
        RETURN 'event'n        ELSE . )      ELSE . ) )
ELSE (     IF ( 'DefectCode Parameter'p = 'DefectCode'n )
    RETURN (       IF ( ' Prim Repl Part Parameter'p Missing )
      RETURN (         IF ( 'event_flag'n = 1 )
        RETURN 'event'n        ELSE . )
      ELSE (         IF ( ' Prim Repl Part Parameter'p = ' Prim Repl Part'n )
        RETURN (           IF ( 'event_flag'n = 1 )
          RETURN 'event'n          ELSE . )       ELSE . ) )    ELSE . ) )
ELSE (   IF ( 'GroupCode Parameter'p = 'GroupCode'n )
  RETURN (     IF ( 'DefectCode Parameter'p Missing )
    RETURN (       IF ( ' Prim Repl Part Parameter'p Missing )
      RETURN (         IF ( 'event_flag'n = 1 )
        RETURN 'event'n        ELSE . )      ELSE (
        IF ( ' Prim Repl Part Parameter'p = ' Prim Repl Part'n )
        RETURN (           IF ( 'event_flag'n = 1 )
          RETURN 'event'n          ELSE . )       ELSE . ) )    ELSE (
      IF ( 'DefectCode Parameter'p = 'DefectCode'n )
      RETURN (         IF ( ' Prim Repl Part Parameter'p Missing )
        RETURN (           IF ( 'event_flag'n = 1 )
          RETURN 'event'n          ELSE . )       ELSE (
          IF ( ' Prim Repl Part Parameter'p = ' Prim Repl Part'n )
          RETURN (             IF ( 'event_flag'n = 1 )
            RETURN 'event'n            ELSE . )         ELSE . ) )
```

Output 5. Output from a nested If-Else Statement

Visual Mode for creating new calculated item:



Display 3. Visual Mode for creating new calculated item.

Note: The attributes of Manufactured and Sold like Model Type, Variant etc. can be directly interacted with the charts/graphs as they will naturally filter both Events and MnS Nos.

ORGANIZATION SPECIFIC REQUIREMENTS

The users had a number of specific requirements for the dashboard design. However, some of the functionalities that they needed aren't directly available in SAS® Visual Analytics. We made extensive and innovative use of the parameters and calculated items to fulfil them. Those requirements and the approaches we used to deploy them are as follows:

DOTTED LINE FOR IMMATURE PRODUCTION BATCHES

In Trend charts, the trend line for those Prod. months that have not yet matured, should be dotted to denote the same. Business understanding was that 95% of the vehicles produced in a month are sold within next 3 months. So, for a Prod. batch to be matured for 0 TIS it should be at least 3 months old. E.g. The Batch of Feb-16 is said to be matured in Jun-16 (T+3Months + TIS = Feb + 3+ 0) for 0 TIS (meaning that we can assume that we will not receive any further events under 0 TIS from this batch). Similarly, the same batch will mature in Sep-16(Feb+3 + 3) for 3 TIS.

In other words, if we are checking the report in Sep 2016, all the prod. batches after Feb-16 should be showed as immature for 3 TIS (the trend line should be dotted) as we can assume that there will be more events coming in from these batches under 3 TIS.

Display 4. Dotted Trend line for immature production batches.

Challenge

No display rule can be used for this purpose as:

a. Display Rules are not dynamic in nature. Every month a new batch becomes matured and the trend line should become solid for that month.
b. Applying different display rules for different MISs is not possible.

Solution

We split the measure (Event) in to 2 parts – Event1 and Event2.

We created a custom Line chart with 2 measures (Event1 and Event2), a shared category (Prod. Month) and a shared group variable (TIS)

We calculated Event1 in a way that it would return missing values for months that are matured for a given TIS.

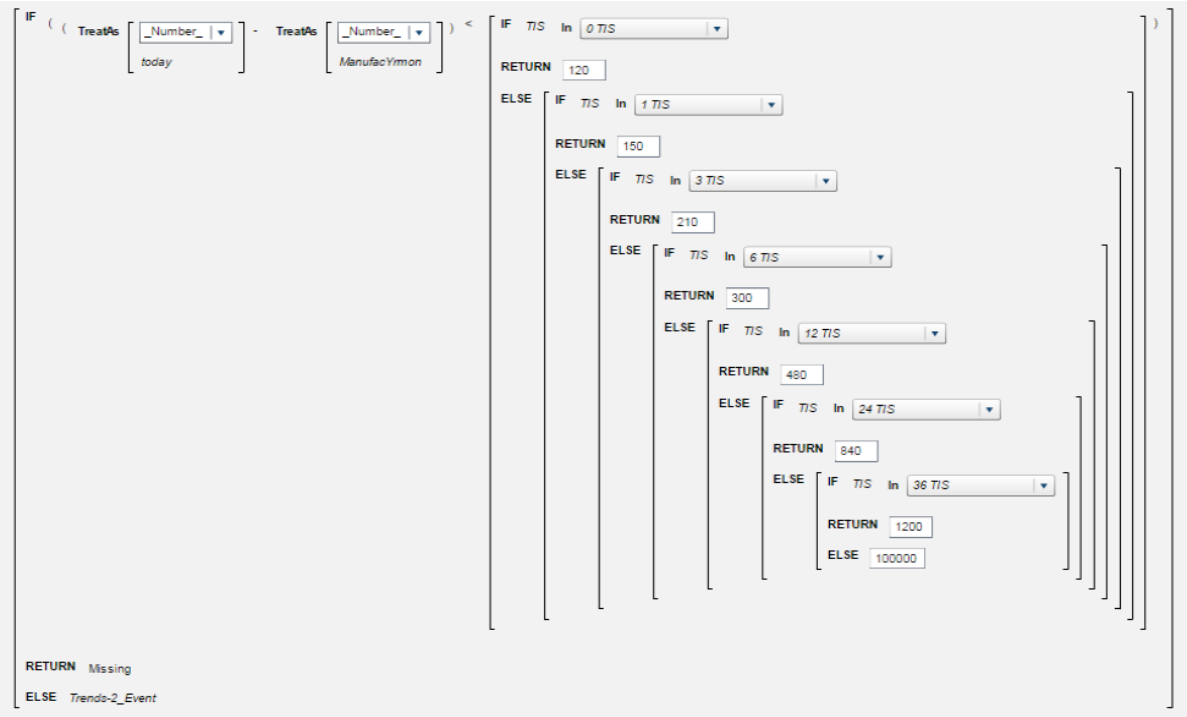
We assigned this Event1 to the dotted line (it would only show the trend-line for the **immature** batches)

We calculated Event2in a way that it would return missing values for months that have **NOT** matured for a given TIS.

We assigned this Event1 to the solid line (it would only show the trend-line for the **mature** batches).

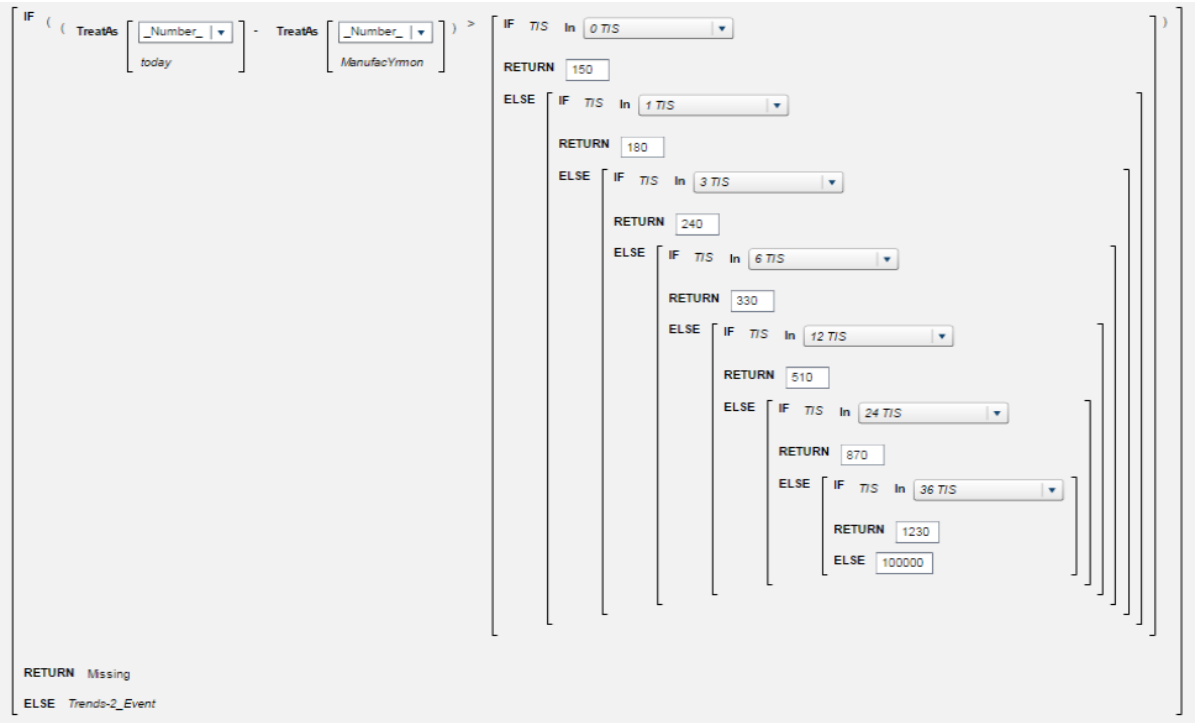
So, for those months for which Event1 is “missing”, Event2will be “not missing” and Vice-Versa.

Event1



Display 5. Visual Mode for creating new calculated item Event1

Event2



Display 6. Visual Mode for creating new calculated item Event2

Note: Event1 and Event2 are calculated according to the difference between today and the production Months.

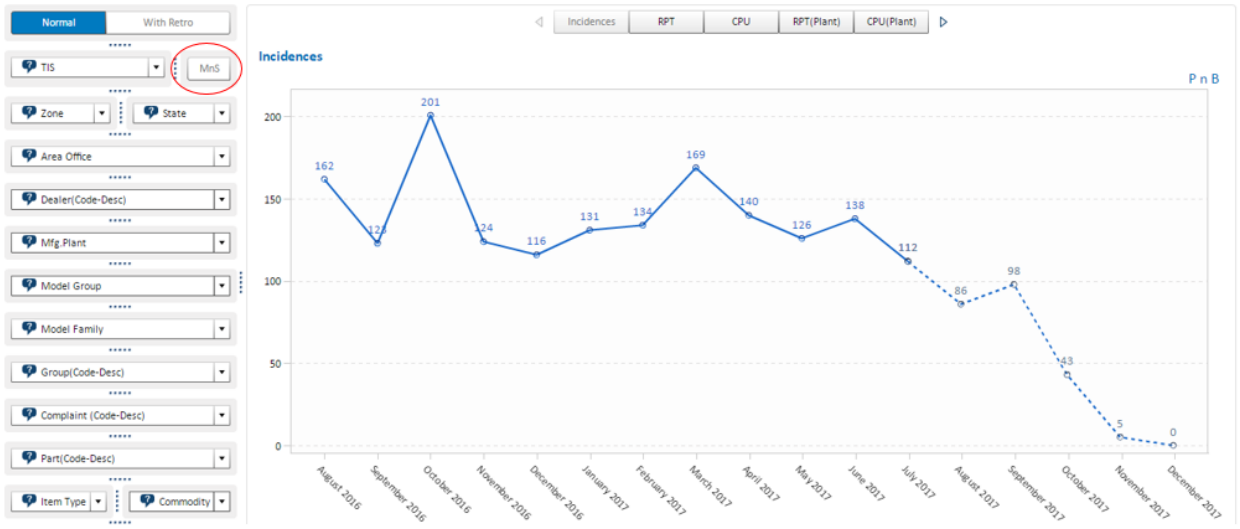
MNS NUMBERS TREND LINE TO BE DISPLAYED ON CLICK OF A BUTTON

Users wanted to be able to compare the Events from a particular production month with the number of vehicles billed from that batch.

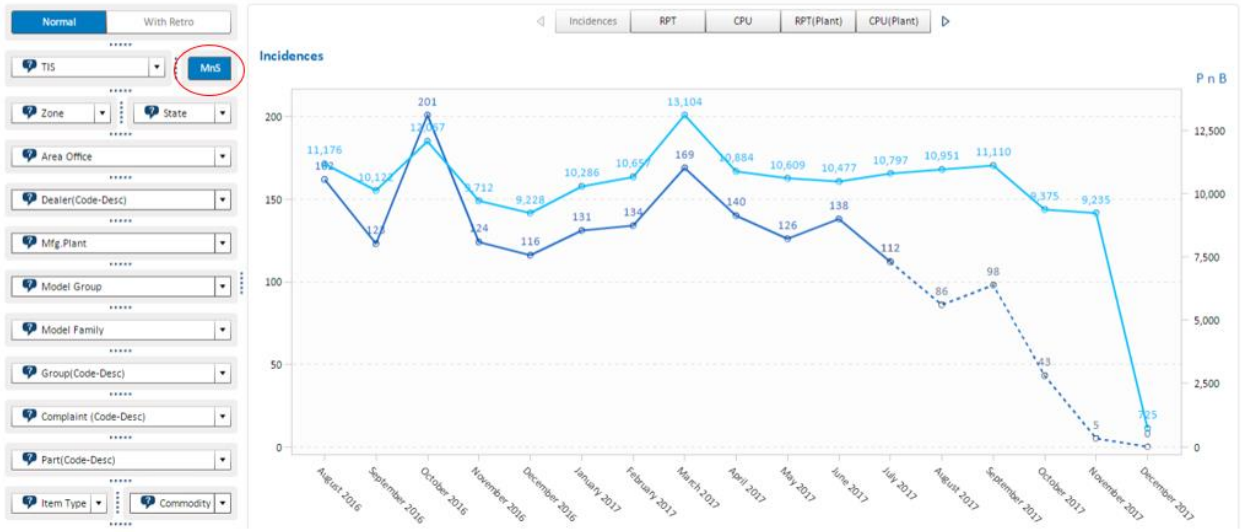
They wanted to be able to populate the MnS trend line on click of a button on the same graph on which events were plotted. And if the button is un-clicked or un-selected, the line should disappear.

Challenge

There is no such functionality available in SAS® Visual Analytics. We can use a Dual-Line Chart, but we need to determine how to make the MnS trend line appear and disappear on click of a Button?



Display 7. Trend line for Event with MnS button unclicked.

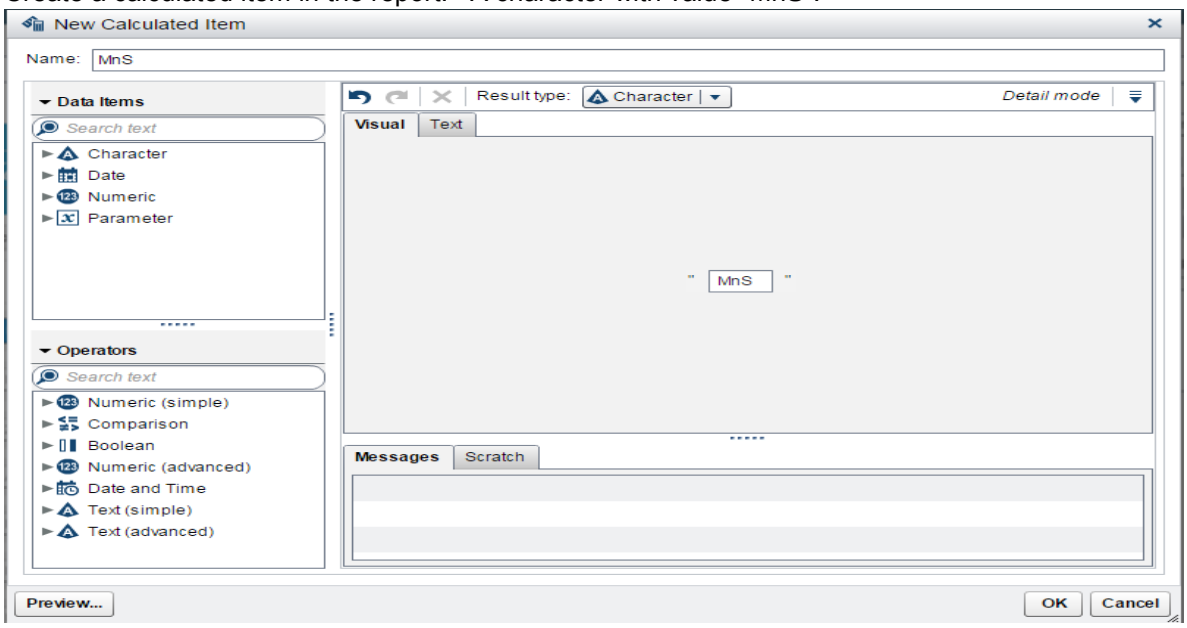


Display 8. Trend line for MnS nos. displayed on click of a button.

Solution

The solution again lies in Parameters and Calculated Items.

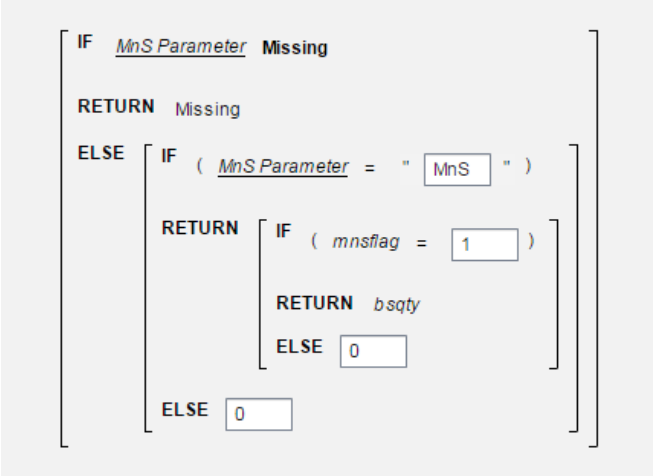
1. Create a calculated item in the report. A character with value “MnS”.



Display 9. New calculated item for displaying ‘MnS’ on MnS button.

2. Assign this item to a button Bar and create a Character Parameter on the Button.

3. Calculate the MnS nos. in a way that only if the parameter contains the value “MnS”, the system should return the billed quantity. If the parameter is missing i.e. there is no selection in the button bar, it should return missing values.



Display 10. Logic for Trend line show hide on click of MnS button.

This way, when the button that says “MnS” is clicked, it will return the MnS values and when the button is un-clicked, it would only return missing values and the trend line would disappear!

SUMMARY

Calculated items in combination with parameters enable us to create a number of functionalities that aren't directly available in SAS Visual Analytics 7.3 or earlier versions. These functionalities/techniques can be applied in analytics across industries for different purposes and requirements where the value of a KPI depends upon the user's selection which makes SAS Visual Analytics a strong analytical tool that can be used for 'on-the-fly' reporting and analysis of data.

REFERENCES

SAS Visual Analytics, User Guide 7.1
SAS Visual Analytics, User Guide 7.3

CONTACT INFORMATION

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

Prateek Kumar Singh
Bristlecone India Ltd.
2nd Floor, Techniplex 1, Techniplex Complex,
Opp. V.S Flyover, Goregaon (W),
Mumbai 62, Maharashtra, India
Work: +91-9987990512
Email: psingh0930@gmail.com

Paresh Peter Rodrigues
Bristlecone India Ltd.
2nd Floor, Techniplex 1, Techniplex Complex,
Opp. V.S Flyover, Goregaon (W),
Mumbai 62, Maharashtra, India
Work: +91-7219663460
Email: paresh590@gmail.com

Saket Shubham
Bristlecone India Ltd.
2nd Floor, Techniplex 1, Techniplex Complex,
Opp. V.S Flyover, Goregaon (W),
Mumbai 62, Maharashtra, India
Work: +91-8348522381
Email: saket.shubham@gmail.com

Shiv Kumar
Mahindra & Mahindra, Information Insight Center (IIC).
4th Floor, Mahindra Towers,
Kandivali East, Mumbai,
Mumbai 400101, Maharashtra, India
Work: +91-8451999852
Email: kumar.shiv7@mahindra.com