Paper 2749-2018

Need Dynamic Interactions in Your Dashboards? Use Parameters!

Abhilasha Tiwari, Accenture Netherlands

ABSTRACT

SAS® Visual Analytics is a robust in-memory environment for creating stunning interactive and dynamic reports. The quick summarization of key performance indicators and the option to navigate the dashboard help in governance and steering operations. Interactive reporting is much more accessible and fun when combined with the use of parameters. Parameters are variables whose value can be changed and referenced by any other object in the reports. When the value of the parameter changes, any report objects that reference the parameter detect the change accordingly. Parameters not only make the dashboards dynamic, but they also make the dashboards intuitive and handy for users. This paper covers how to use parameters to view your data dynamically in a graph or table over a specified span of days.

INTRODUCTION

Parameters in SAS® Visual Analytics is a robust functionality and provide users the option to regulate the modification and interaction with the report. Parameters are available for controls in reports which in turn give flexibility to the complex dashboards. Once a parameter is created and assigned to a control in a report, it can also be used in complex calculations, filters, ranks, and display rule. Any change in the value of the parameter is reflected on the report objects that reference the parameter, and gets updated.

In this paper is discussed, different types of parameters and controls supporting the parameters in the report. Also, explained with examples the functionality in filtering the data dynamically in the tables/graphs of the report.

TYPES OF PARAMETERS

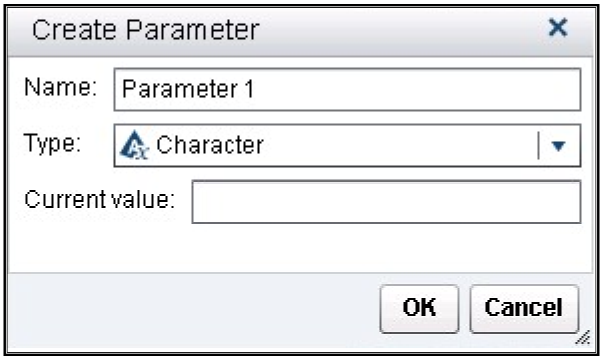
Parameters in SAS® Visual Analytics are of two types:

1) **Character Parameter :**

It accepts the only character as the value.

Not possible to provide start and end value.

Name of the parameter can be changed.



Create Parameter

Name: Parameter 1

Type: Character

Current value:

Figure 1. Character Parameter

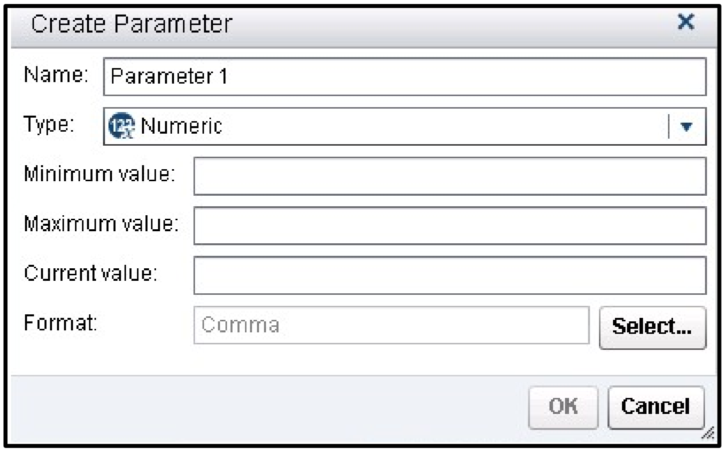
2) **Numeric Parameter:**

It accepts only numeric as the value.

Providing range i.e. minimum and maximum value is possible.

Name of the parameter can be changed.

The format of the value can be changed.



Name: Parameter 1

Type: Numeric

Minimum value:

Maximum value:

Current value:

Format: Comma

Figure 2. Numeric Parameter

CONTROLS AND FEATURES SUPPORTING PARAMETERS

Parameters can be used in filters, ranks, calculations and display rules.

Controls supporting parameters:

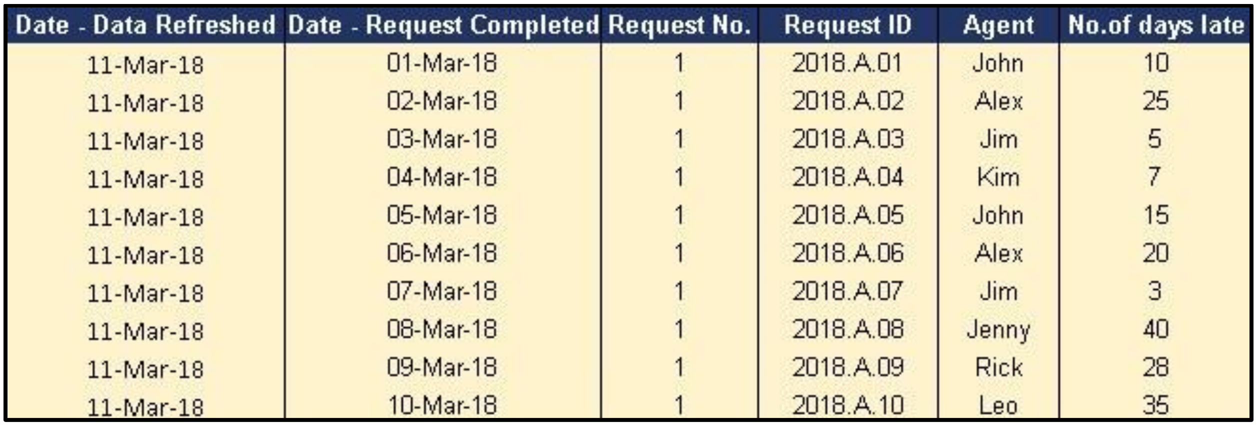
1) **Text Input:** Supports both character and numeric parameter
2) **Button Bar:** Supports character parameter only
3) **Drop-Down list:** Supports character parameter only
4) **Slider:** Supports numeric parameter when the range in the slider is a single point.

Features supporting parameters:

1) **Calculations**: Calculating new items and aggregated measure supports the use of parameters. It can be a character or numeric depending on the requirement
2) **Display Rules**: Numeric parameters could be specified as the value of an expression rule
3) **Filters**: Supports both character and numeric parameters depending on the need of the report
4) **Ranks**: The parameter can be included for the n value of the rank. It is supported for Top Count, Bottom Count, Top Percent and Bottom Percent.
5) **URL’s**: Supports both character and numeric parameters and could be used to provide the user specific URL links for the reports

EXAMPLE DATASET

A mock-up dataset is created to be used in this paper to explain the dynamic functionality of the parameters.



Date - Data Refreshed	Date - Request Completed	Request No.	Request ID	Agent	No.of days late
11-Mar-18	01-Mar-18	1	2018.A.01	John	10
11-Mar-18	02-Mar-18	1	2018.A.02	Alex	25
11-Mar-18	03-Mar-18	1	2018.A.03	Jim	5
11-Mar-18	04-Mar-18	1	2018.A.04	Kim	7
11-Mar-18	05-Mar-18	1	2018.A.05	John	15
11-Mar-18	06-Mar-18	1	2018.A.06	Alex	20
11-Mar-18	07-Mar-18	1	2018.A.07	Jim	3
11-Mar-18	08-Mar-18	1	2018.A.08	Jenny	40
11-Mar-18	09-Mar-18	1	2018.A.09	Rick	28
11-Mar-18	10-Mar-18	1	2018.A.10	Leo	35

Figure 3. Mockup dataset

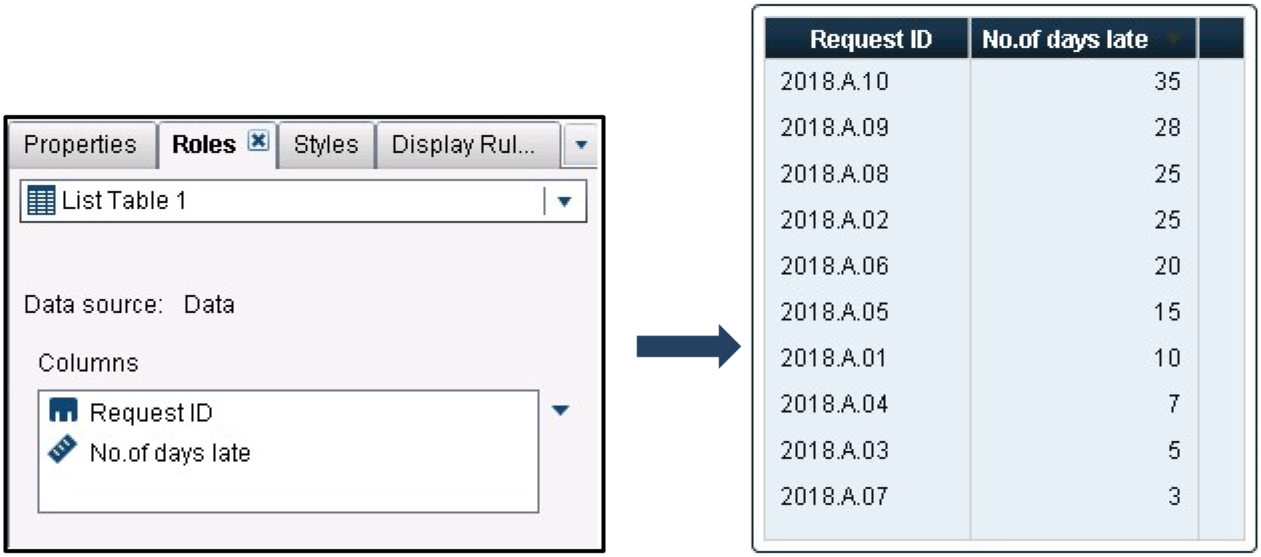
This mockup dataset represents the data from a call center where *Requests* from customers are received and registered by the *Agents*.

- **Date-Data Refreshed:** Refers to the date when the data was refreshed for the last time
- **Date-Request Completed:** Refers to the date when the request was completed
- **Request No.:** Refers to the number of requests
- **Request ID:** Refers to the unique id of each request made by the customer
- **No. Of days late:** Refers to the days that the request was completed and late for their scheduled time.

CASE 1: VIEW REQUESTS CLOSED MORE THAN X DAYS PAST SCHEDULED TIME

The requirement here is to view all the *Requests* that were completed and late by 'X' days. 'X' here refers to the number of days as per user's choice. It could be three days, ten days, thirty days, etc.

- Select 'List Table' from *Objects Pane* and drag it to the workspace. Under the *Roles Tab*, add the values 'Request ID' and 'No. of days late'.



Data source: Data

Columns

Request ID

No.of days late

Request ID	No.of days late
2018.A.10	35
2018.A.09	28
2018.A.08	25
2018.A.02	25
2018.A.06	20
2018.A.05	15
2018.A.01	10
2018.A.04	7
2018.A.03	5
2018.A.07	3

Figure 4. Adding Values under Roles Tab

The table displays the data for all the requests that are completed and are late from their due date.

- Select and drop the 'Text input' from *Objects pane* into workspace and right click to 'Create Parameter.'

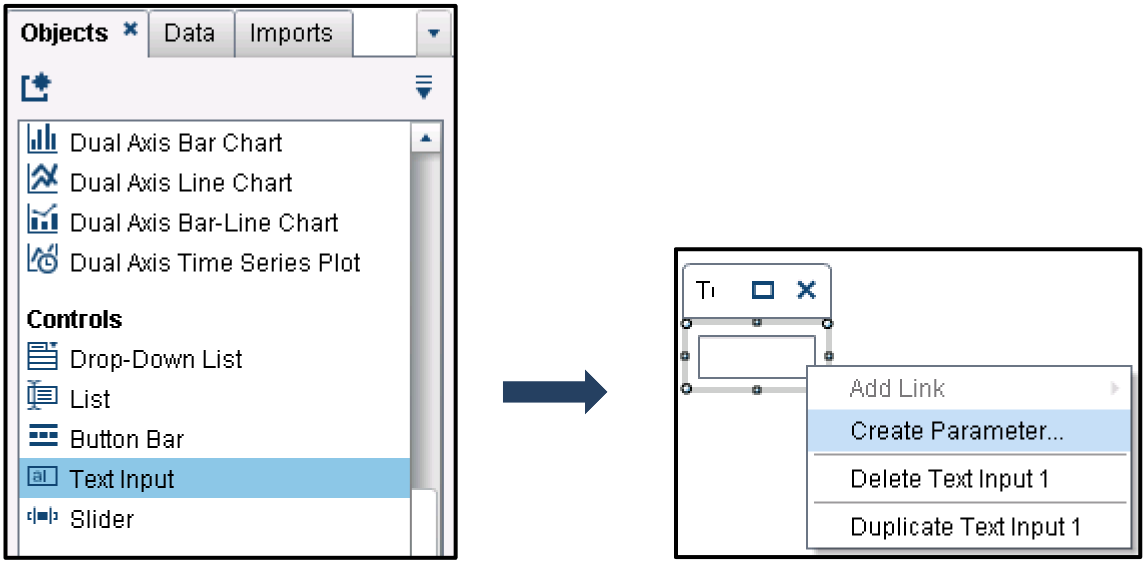
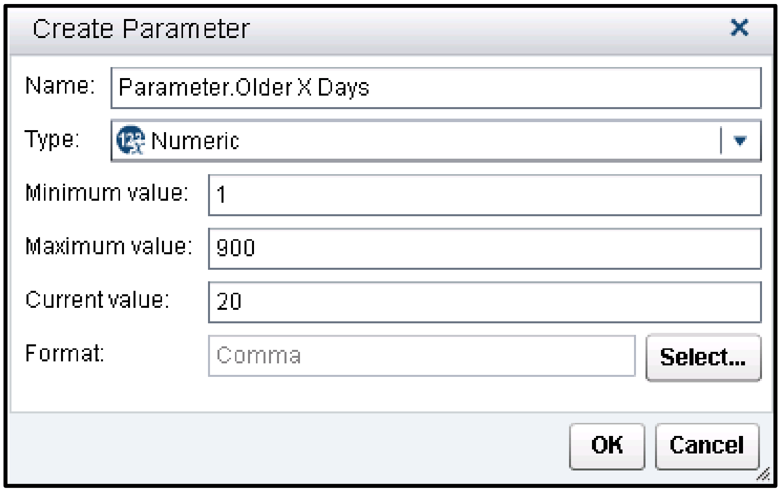


Figure 5. Creating Parameter

- A new window is opened to add the details for the parameter.
    - The parameter name is changed to ‘Parameter.OlderXDays'
    - The minimum value is 1, and the maximum value is 900 (a case can only be late by at least one day, and it is reasonable to assume that no cases are late more than 900 days).
    - The current value is specified as 20 to view all the requests that were completed late by 20 days.
    - After entering the values for the parameter, click ‘Ok.'


Create Parameter

Name: Parameter.Older X Days

Type: Numeric

Minimum value: 1

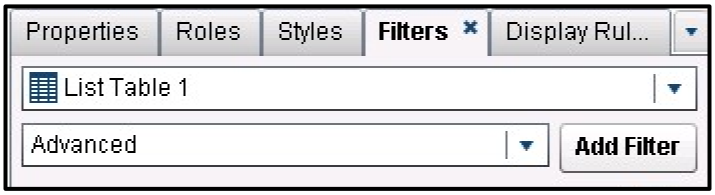
Maximum value: 900

Current value: 20

Format: Comma

Figure 6. Entering values for the parameter

- Select the Table and go to *Filters Pane* to apply condition in the *Advanced Filter* for the parameter to be referenced by the table.

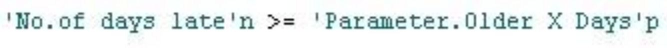


List Table 1

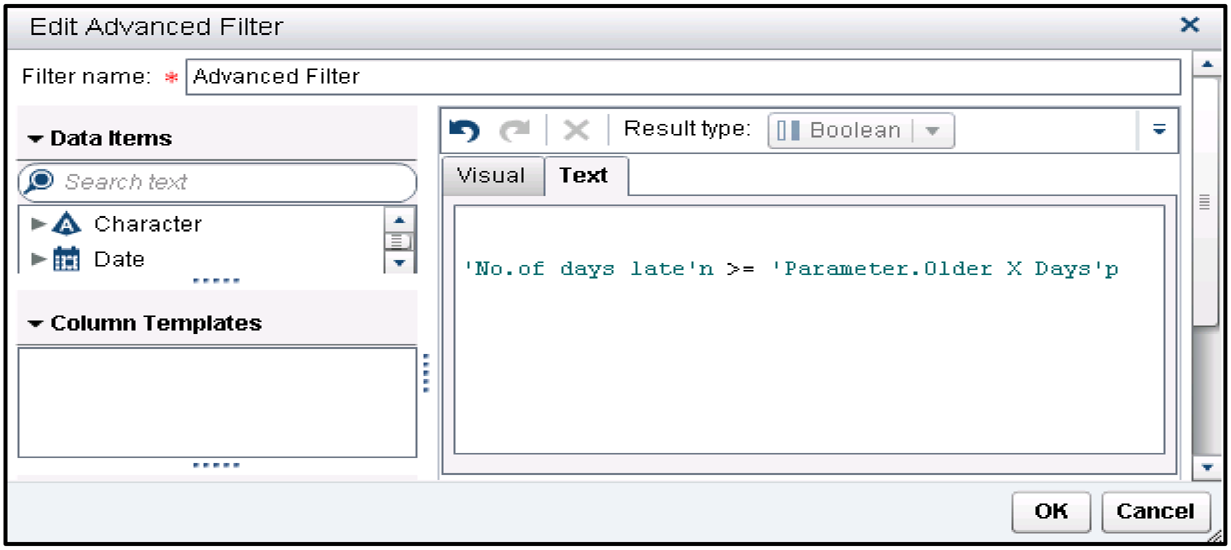
Advanced

Figure 7. Adding Advanced Filter

- Add a condition in the filter where the value of ‘No. of days late’ is more and equal to the value entered in the Parameter.



```
'No.of days late'n >= 'Parameter.Older X Days'p
```



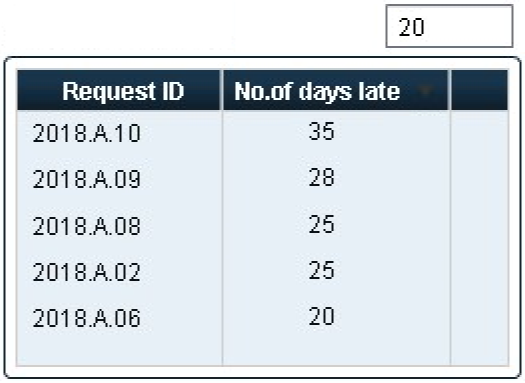
Filter name: * Advanced Filter

Result type: Boolean

```
'No.of days late'n >= 'Parameter.Older X Days'p
```

Figure 8. The condition under Text in the Advanced filter

- Once the condition is applied, the table is filtered according to value in the parameter (in this example 20). The table displays the five requests that are completed late by 20 or more than 20 days.

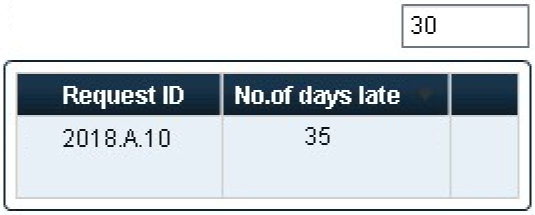


20

Request ID	No.of days late
2018.A.10	35
2018.A.09	28
2018.A.08	25
2018.A.02	25
2018.A.06	20

Figure 9. Data in the table is updated as per parameter value

- Select the table and change the value in the parameter of the table to 30.The table gets updated and displays one request that is completed late by 30 or more than 30 days.



30

Request ID	No.of days late
2018.A.10	35

Figure 10. Data in the table is updated as per parameter value

Similarly, the value of the parameter can be changed to any number of days and data can be filtered/selected as per users choice dynamically in the reports.

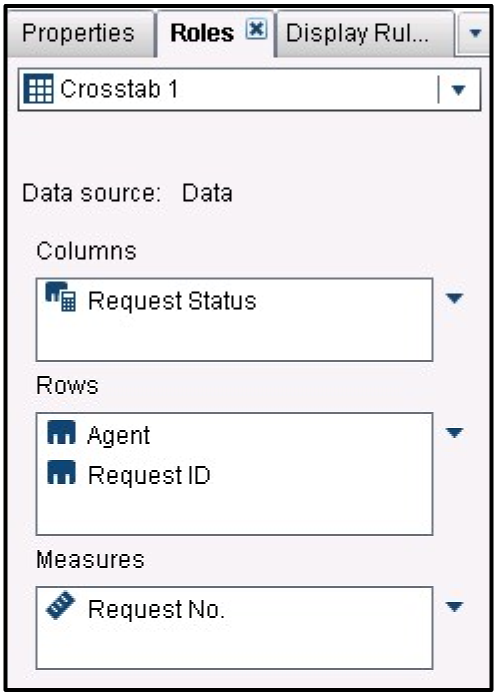
CASE 2: VIEW NUMBER OF REQUESTS THAT ARE COMPLETED LATE WITH THE AGENT INFORMATION

Now, let's have a look at the same information mentioned in **Case 1**, but on agent level. The requirement here is to view the number of *Requests* handled by *Agent* that were completed and late by at least 'X' days. 'X' here refers to the choice of the user. It could be three days, ten days, 30 days, etc.

- Select 'Cross Tab' from *Objects Pane* and drag it to the workspace. Under the *Roles Tab*, add the values for 'Agent', 'Request ID' under *Rows* and 'Request No.' under *Measures*.
- Calculate a new item 'Request Status' using parameters (mentioned in Case1) where status is 'Late' if the value of the variable 'No. of days late' is not between 0 and value entered in the parameter, else it is 'not late by X days.'

```
IF ( ( 'No.of days late'n NotMissing ) AND ( 'No.of days late'n
NotBetweenInclusive(0, 'Parameter.Older X Days'p) ) )
RETURN 'Late'
ELSE 'not late by X days'
```

- Use this new calculated item 'Request Status' in the cross tab under *Column* section.



Crosstab 1

Data source: Data

Columns

Request Status

Rows

Agent

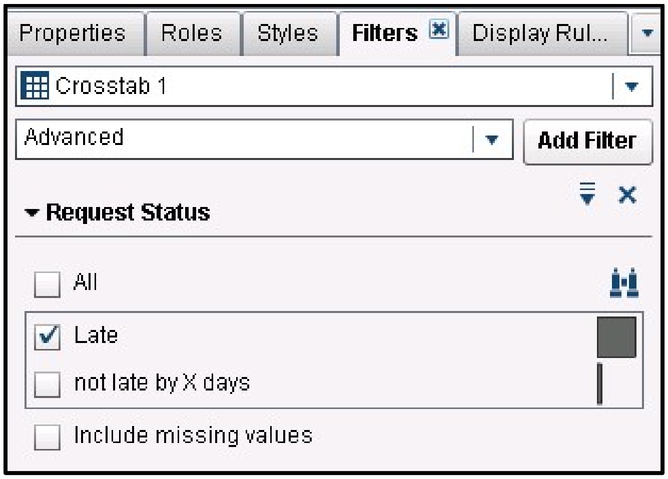
Request ID

Measures

Request No.

Figure 11. Adding Values under Roles Tab

- Select the cross tab and apply a filter for ‘Request Status’ to show only values for the requests that are late by X days.



Crosstab 1

Advanced

Request Status

□ All

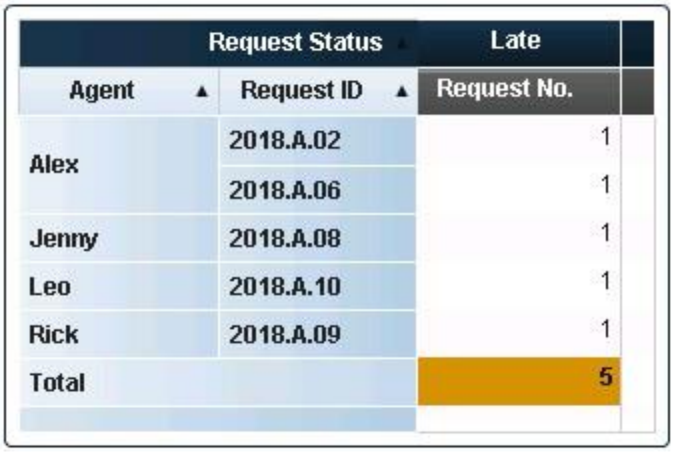
☑ Late

□ not late by X days

□ Include missing values

Figure 12. Filter for Request Status

- Change the values in the parameter to 20. The table gets updated, and data are shown for all the requests that were completed late by 20 or more days along with agent information.



	Request Status	Late
Agent	Request ID	Request No.
Alex	2018.A.02	1
	2018.A.06	1
Jenny	2018.A.08	1
Leo	2018.A.10	1
Rick	2018.A.09	1
Total		5

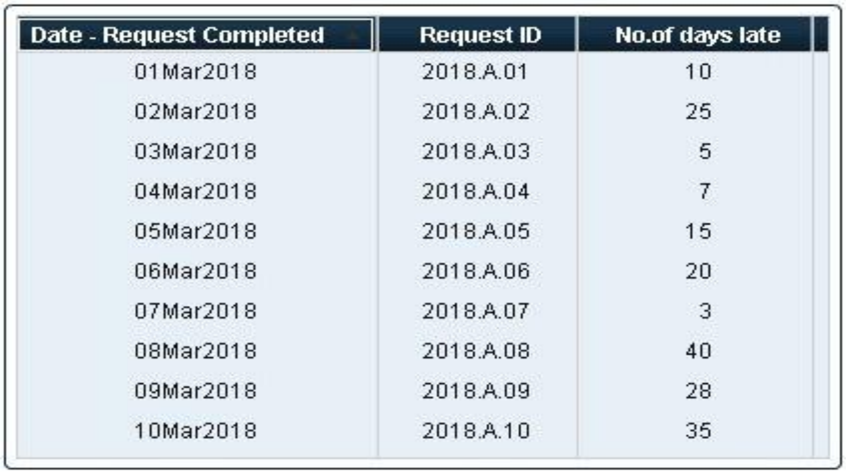
Figure 13. Data in the table is updated as per parameter value

From the table above, the information about agent Alex is displayed that he completed late two requests 2018.A.02 and 2018.A.06 respectively. Similarly, Jenny, Leo, and Rick completed late one request 2018.A.08, 2018.A.10, 2018.A.09 respectively.

CASE 3: VIEW REQUESTS COMPLETED IN PAST X DAYS

The requirement here is to view the number of the *Requests* over the period of 'X' days. 'X' here refers to the choice of days by the user. It could be three days, ten days, 30 days, etc.

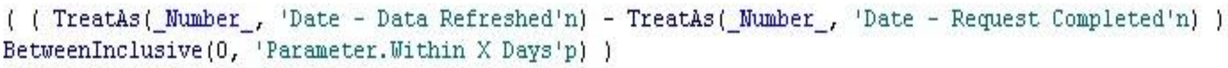
- Select 'Tables' from *Objects Pane* and drag it to the workspace. Under the *Roles Tab*, add the values for 'Date', 'Request Type' and 'No. of days late'.



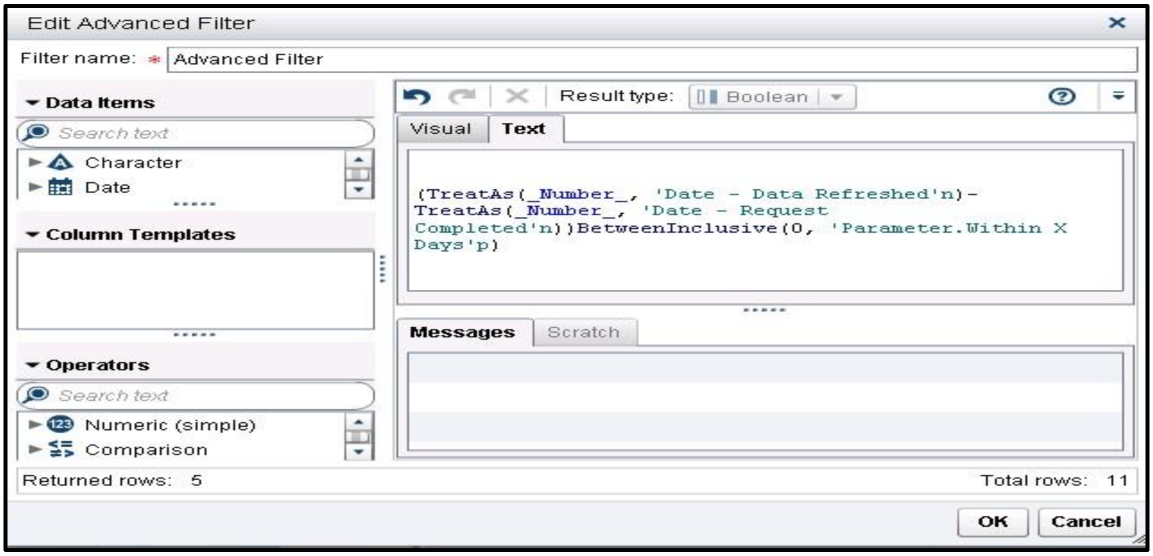
Date - Request Completed	Request ID	No.of days late
01Mar2018	2018.A.01	10
02Mar2018	2018.A.02	25
03Mar2018	2018.A.03	5
04Mar2018	2018.A.04	7
05Mar2018	2018.A.05	15
06Mar2018	2018.A.06	20
07Mar2018	2018.A.07	3
08Mar2018	2018.A.08	40
09Mar2018	2018.A.09	28
10Mar2018	2018.A.10	35

Figure 14. Values in List Table

- Create a numeric parameter as mentioned in **Case 1** and name parameter as 'Parameter.Within X Days'.
- Add a condition in the advanced filter where the value of the dates is within the range of value entered in the parameter.To achieve the above, following points needs to be considered
    - The date should be converted to a number using the TreatAs function.
    - Date as of today or date of data refreshment should be taken into account for calculating the number of days.
      Both of the dates should be converted into numeric using the TreatAs function.
      In below example, date of data refreshment (11th March 2018) is considered.
    - Condition applied:
      (TreatAs (_Number_,Date-Data Refreshed) - TreatAs(_Number_,Date-Request Completed)) BetweenInclusive (0, Parameter.Witihin X Days))



```
( ( TreatAs(_Number_, 'Date - Data Refreshed'n) - TreatAs(_Number_, 'Date - Request Completed'n) )
BetweenInclusive(0, 'Parameter.Within X Days'p) )
```



Filter name: * Advanced Filter

Result type: Boolean

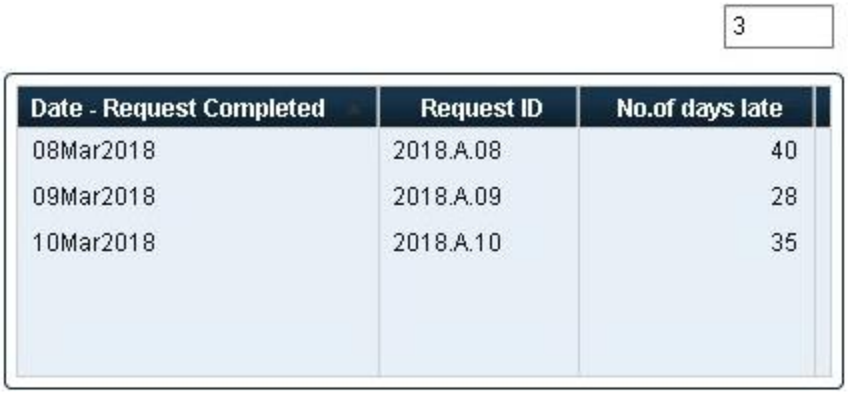
```
(TreatAs(_Number_, 'Date - Data Refreshed'n)-
TreatAs(_Number_, 'Date - Request
Completed'n))BetweenInclusive(0, 'Parameter.Within X
Days'p)
```

Returned rows: 5

Total rows: 11

Figure 15. Advanced Filter with the condition

- Once the condition is applied, select the table and change the value in the parameter of the table to three.



3

Date - Request Completed	Request ID	No.of days late
08Mar2018	2018.A.08	40
09Mar2018	2018.A.09	28
10Mar2018	2018.A.10	35

Figure 16. Data in the table is updated as per parameter value

The data in the table gets updated and shows all the requests that were completed late from past three days, since the data was refreshed, i.e. 11th March 2018.

Thus by changing only the value in the parameter, the user can analyze the data from past few days to week and months.

CONCLUSION

Parameters in SAS® Visual Analytics is a robust functionality for creating stunning, interactive and dynamic reports. One of the essential components for effective and efficient reporting is to provide users the opportunity to regulate the information as per their needs. Gone are the days of the static reporting were information is displayed with limited space to find more insights from the data. The use of parameters in the calculation, filters, and ranks not only provide flexibility to the complex dashboards but also allows users to have control on the data ranges in the report.

REFERENCES

SAS Institute Inc. “SAS(R) Visual Analytics 7.1: User's Guide”. Accessed March 18, 2018
http://support.sas.com/documentation/cdl/en/vaug/67500/HTML/default/viewer.htm#p0qvah37e3ft42n1cbdyymfvbbjl.htm

SAS Communities. ‘Date parameters in SAS Visual Analytics’. Accessed March 18, 2018
https://communities.sas.com/t5/SAS-Communities-Library/Date-parameters-in-SAS-Visual-Analytics/ta-p/233338

CONTACT INFORMATION

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

Abhilasha Tiwari
Accenture Netherlands
+31 (0) 610120389
tiwari.abhilasha1@gmail.com