

Spotfire Patient Profile graphs: An Abridged effective approach to better patients data understanding using Scripting Languages

Anish Kuriachen, Syneos Health, New Jersey, USA

Hrudeep Antony, Syneos Health, North Carolina, USA

ABSTRACT

The patient profile plots play a significant role in the monitoring and reporting of subject safety information from the onset of a clinical trial. Traditional patient profile plots are created based on safety data such as lab results, vital signs, drug exposure, etc. The classic patient profile graphs are usually plotted on the same x-axis using the duration of the subject in the study as the x-axis reference range. This approach occasionally leads to illegible graphics output, especially if the study has several time points or visits that collect safety information. In addition to this challenge, the traditional patient profile requires the corresponding listing outputs to interpret the subject safety profile, since graphics by itself may not express the whole story about the subject.

This paper will present the use of TIBCO Spotfire to build the best patient profile with examples. This not only provides the best quality graphics but also a dynamic reporting system that helps with the cross-referencing of subject's data without the need of flipping through the listings.

INTRODUCTION

TIBCO Spotfire is a powerful data analytics tool that lets users create smart, secure, flexible, and scalable Visualizations. The SDTM datasets are created using the SAS software. SDTM data standards have several flags and variables that facilitate the formation of effective patient profile graphics in Spotfire as shown in the flowchart in figure 1.

One of the several advantages of using Spotfire visualization is its dynamic and scalable reporting feature, allowing the users to choose their own criteria for the visualization. The criteria can be changed based on the need of the reviewer. In a clinical trial reporting setting, this feature will allow the users to easily filter for desired information (e.g. different treatment groups, different populations) while exploring the data in Spotfire visualizations. Spotfire, by default, has several options to explore the data but the prominence of using Spotfire is in its ability to further expand these customization features based on the users' ability to develop scripts in languages such as R, Python, or S-PLUS.. This feature contributes to the overall efficiency of reviewing and exploring the data. Some of the customization techniques using R and Python scripts are further discussed in this paper.

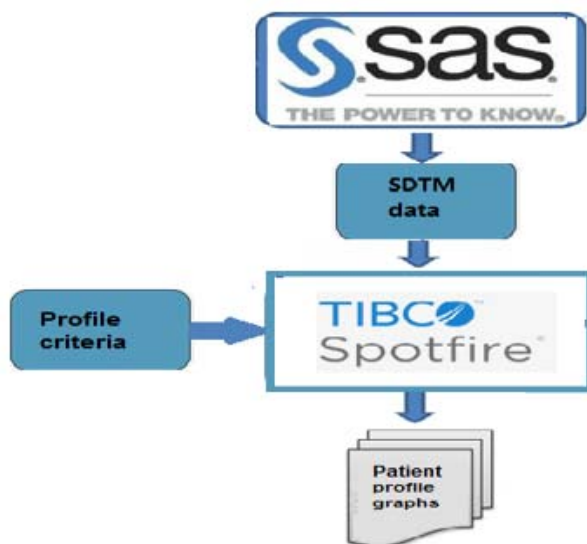


Figure1: Process flow chart

CREATING VISUALISATION PACKAGES

The first step in creating Spotfire Visualization packages is to load the SDTM datasets to Spotfire by selecting the "File-Add Data Tables" on the menu and navigating to the corresponding folders. Spotfire allows users to process the database by allowing the users to add new variables defined from the existing data column, transpose the data, combine multiple datasets, etc. Using the loaded and processed data the user can generate listings, tables, and graphs according to the requirements.

Some of the techniques to create effective patient profile plots are further discussed in the examples below.

EXAMPLE 1

In the first example, a profile plot for LAB data of maximum ALT (Alanine Aminotransferase) value vs. maximum Bilirubin value for each subject is created. The x-axis and the y-axis parameters are selected using the script as shown in figure 2 below. The output plot generated is shown in figure 3 below. Note that each subject's information is color-coded and it is easy to analyze the overall trend of the subjects ALT/Bilirubin LAB profile. The individual subject's information and outliers can be conveniently identified and filtered using the filter panel. Further information on the subject can be obtained by merely moving the mouse pointer towards the panel as shown in the example.

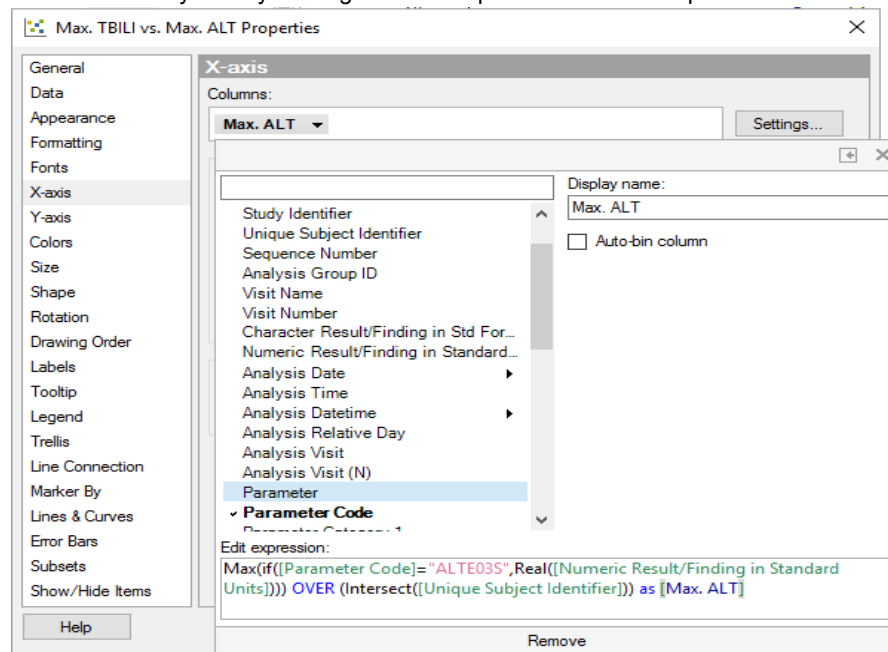


Figure 2: Script to get Maximum ALT value for each subject

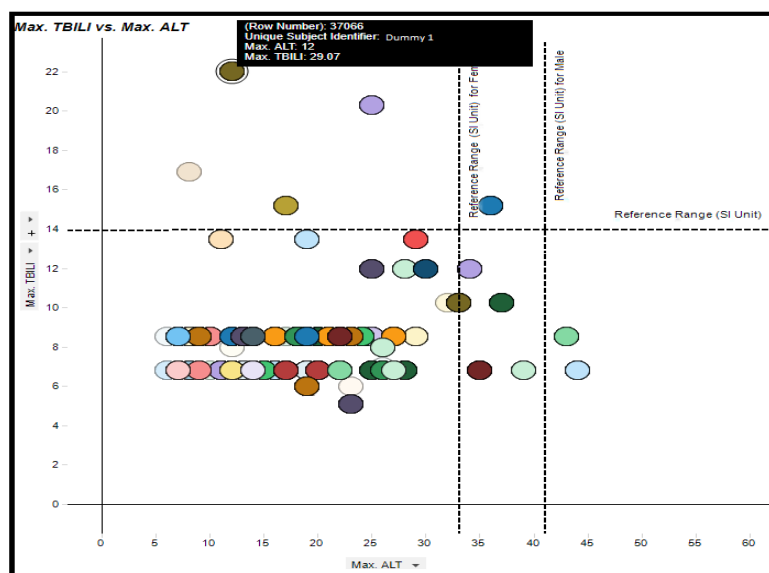


Figure 3: Color-coded visualization of each subject.

EXAMPLE 2

In this example, we are creating a profile plot for maximum ALT/AST value (shown in figure 4) vs. maximum bilirubin value. In this example, we used the same script as in example 1, but with the addition of allowing the user to dynamically choose the x-axis variable. The parameter name is replaced with “\${altast}” which will accept values from the dropdown menu as shown in figure 5 below.

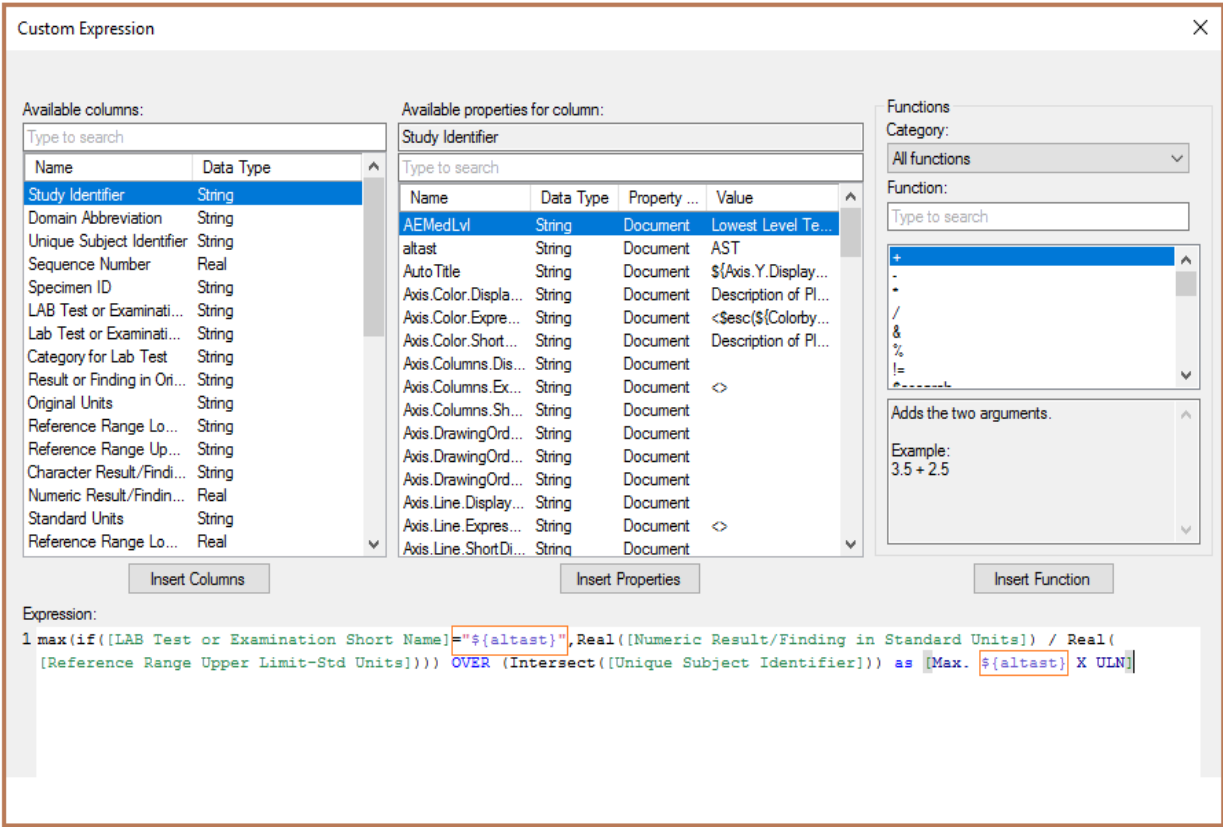


Figure 4: Script to get maximum ALT/AST value for each subject

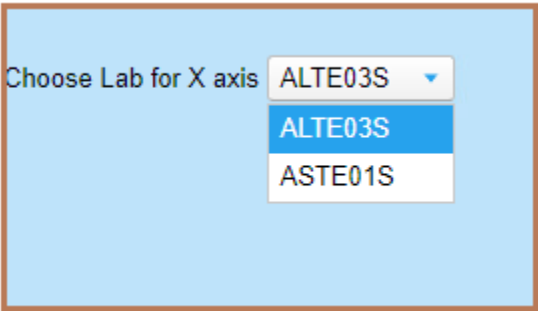


Figure 5: Dropdown menu to choose user-defined x-axis.

The drop-down menu is created using by choosing the “Control type” in the text area menu and by adding “altast” as the new property control as shown in figure 6 below.

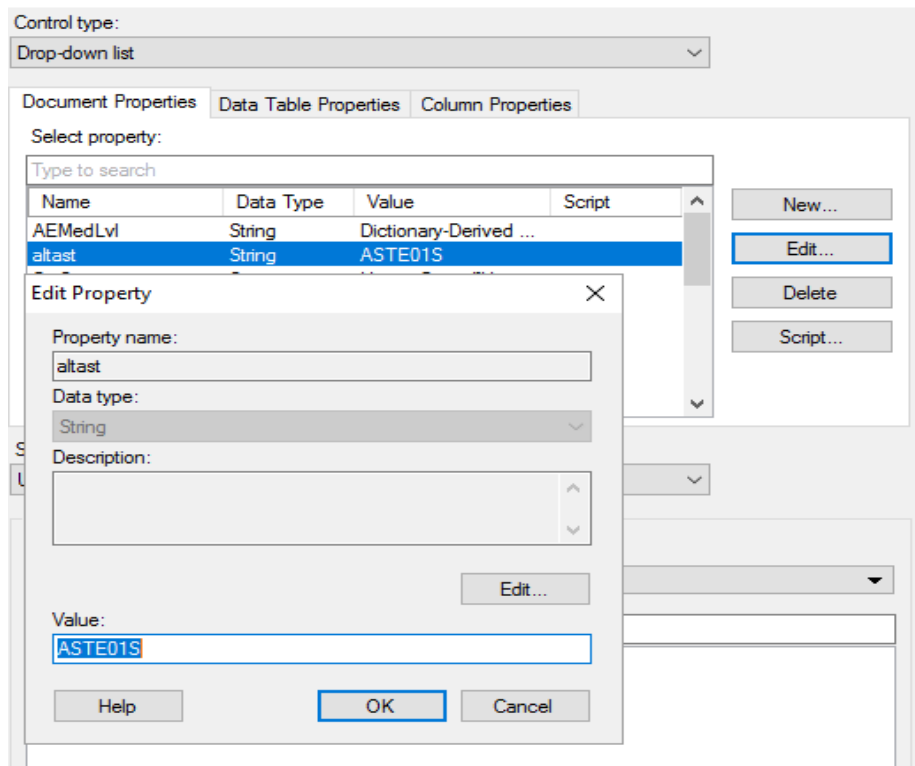


Figure 6: Control type selection window

The Output visualization example using the parameter selection can be seen in figure 7 below. Note that in this example AST parameter has been selected for x-axis to compare with the Bilirubin values.

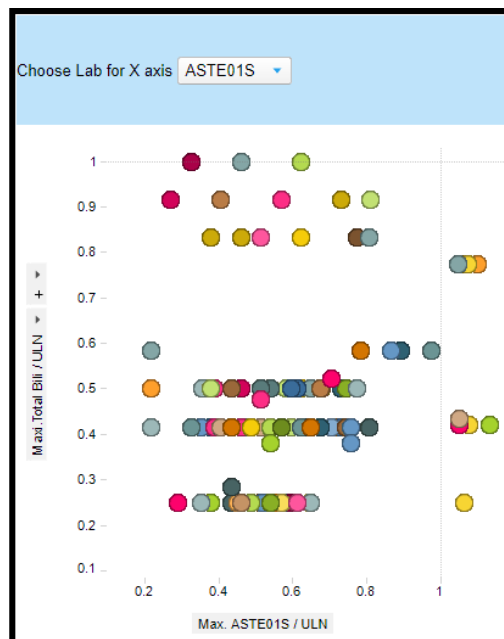


Figure 7: Plot generated using user-defined x-axis

Using a similar methodology user can choose to add any number of parameters such as treatment, race, sex, etc. to the drop-down menu, to compare the results. This parameter addition technique will facilitate reviewers to instantly create their own customized plots and review the results without need for any programming.

EXAMPLE 3

In this example, we are creating a zoom slider in the Spotfire visualization. Zoom sliders are used to get a closer look at details in your visualization by providing the ability to adjust the axis scales. An example of output with zoom sliders can be found in figure 8 below.

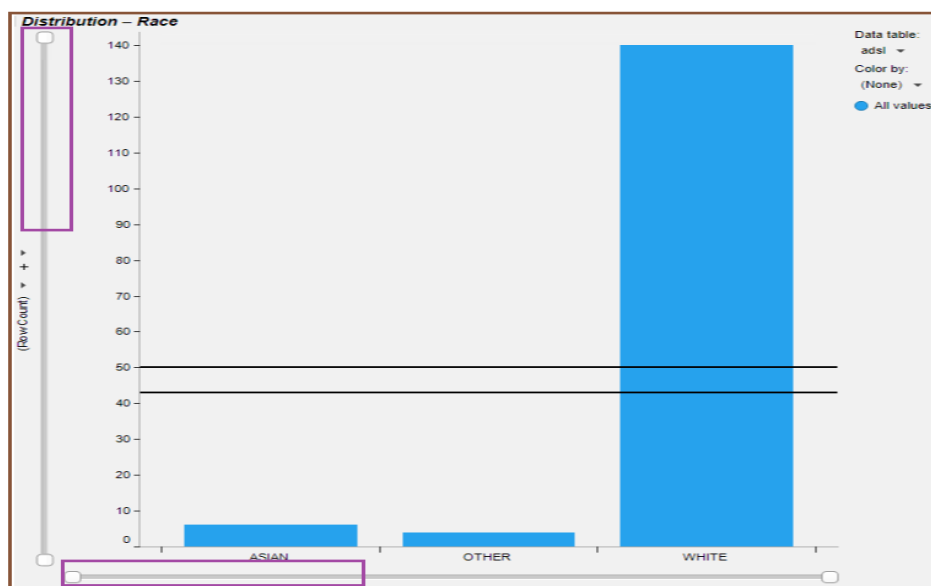


Figure 8: Option to choose zoom slider on/off

The typical way to enable zoom sliders is by clicking on the small arrow icon in the title bar of visualization as shown in figure 9 below. The icons in the title bar are only shown when hovering with the mouse pointer over the title bar area.

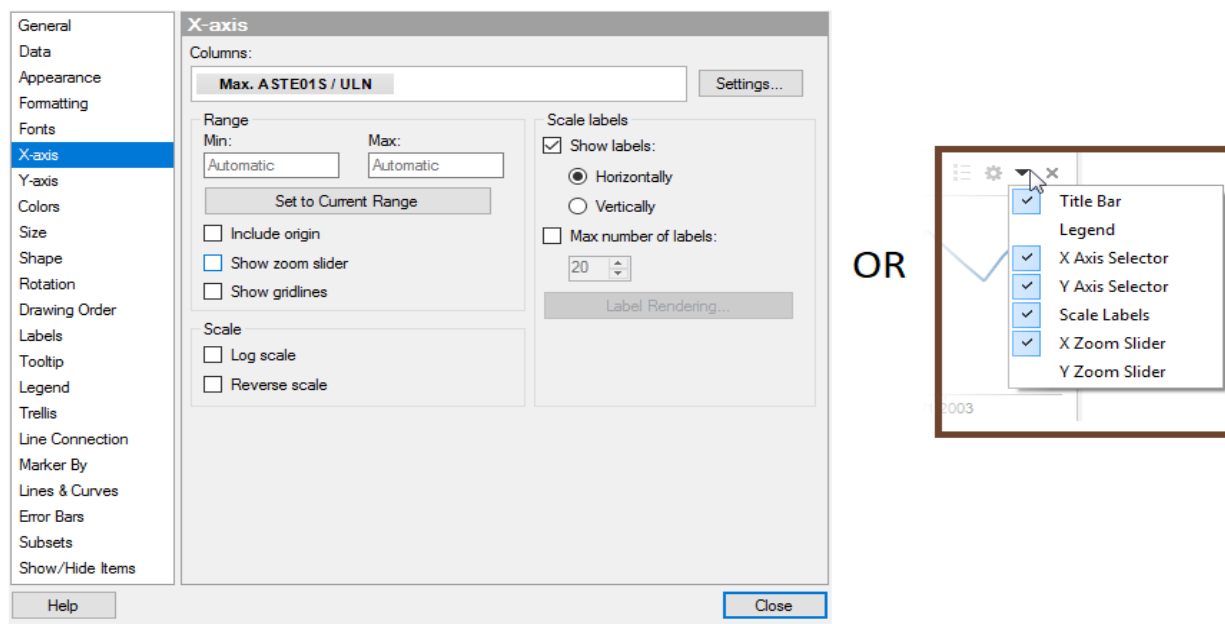


Figure 9: Option to choose zoom slider on/off

If there are numerous visualizations in the package, it may be tedious to add the zoom sliders to each of the plots. In such a situation, another practical way to add the zoom slider is by adding a button that allows the users to enable a zoom slider on/off for the visualization in a page with the use of a python script.

The following steps may be followed to add a zoom slider button with the Python script.

- Step 1: Insert a text area by selecting the “insert” menu and choosing the “text area” option.
- Step 2: Add the “Action Control” item from the text area menu as highlighted in yellow in figure 10 below.
- Step 3: Select “Scripts” from the left tab and choose Control Type as “Button”. Now select the “New” button to add new script for the zoom sider on/off.

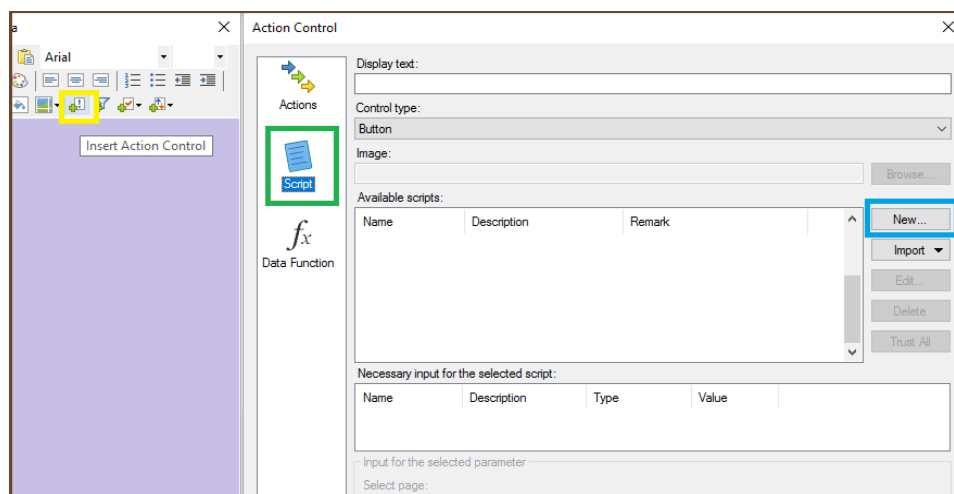


Figure 10: Inserting a Button in a page

- Step 4: Clicking on the “New” button will prompt a pop-up window that allows users to add a script. In this example, the script name is “zoomonoff”. The Python script for adding the zoom slider can be found in figure 11 below. The active scripts that can be used across all the plots in the package are displayed in the window in figure 12 below.

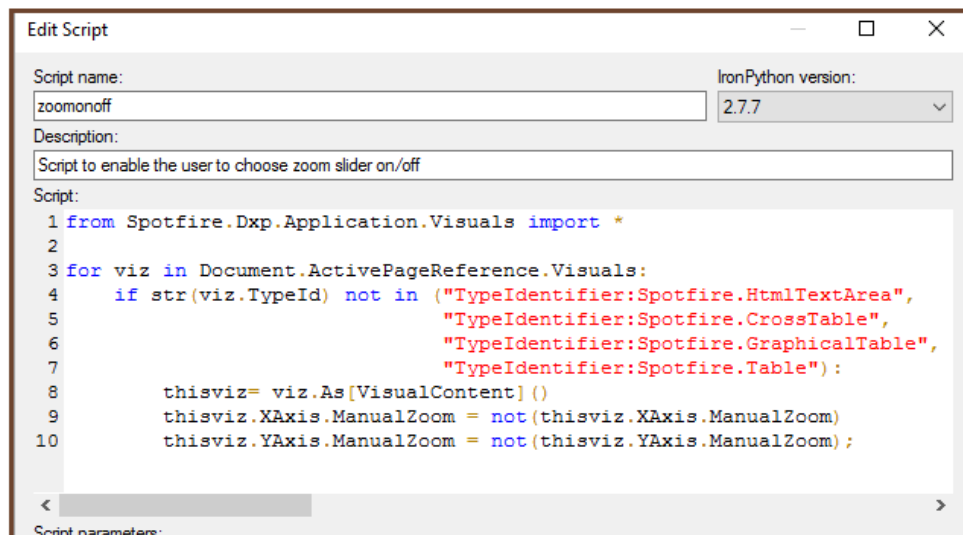


Figure 11: Script for enabling zoom slider in graphs.

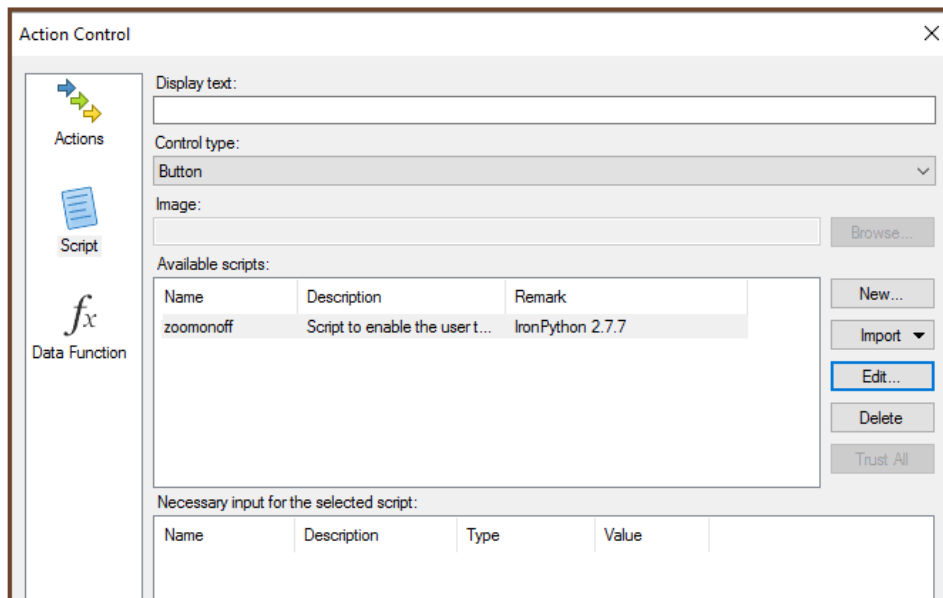
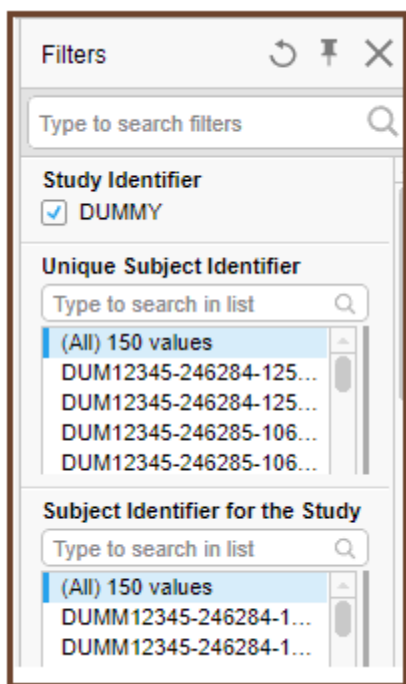


Figure 12: Scripts available in the visualization package

Example 4

All the variables of the data table in the Spotfire can be associated with a filter that can be found on the filter panel. These filters are used to limit the data based on the subset of interest. Depending on the type of data, the filters types will also vary. The Spotfire filter panel and the different types of filters can be found in figure 13 below.



Number	Filter Type
1	Range Filter
2	Item Filter
3	Radio Button Filter
4	Check Box Filter
5	Text Filter
6	List Box Filter
7	Hierarchy Filter

Figure 13: Filter Pannel and Filter Type

When several filters are active, the user may find it difficult to revert back to the original data view especially when the data tables consist of numerous variables. In this example, we will provide an option for resetting all filters in the Spotfire package using a simple Python script as shown in figure 14 below. Using the steps explained in example 3 above, a “Reset button” is created which activates the script and resets all the filters in the panel.

```
Script:
1 import Spotfire.Dxp.Application.Filters as filters
2
3 for scheme in Document.FilteringSchemes:
4     scheme.ResetAllFilters()
```

Figure 14: Script for resetting all filters

Example 5

In this example, we are creating an x-axis range limiter option that allows the user to choose a dynamic axis range. Two text input fields such as minimum and maximum range values are added for the x-axis. The following steps can be followed to add this feature to the visualization.

Step1: Add two text boxes in the text area by selecting “Input Field” from the text in menu and create corresponding two new variables to store the text area value as shown in figure 15 below.

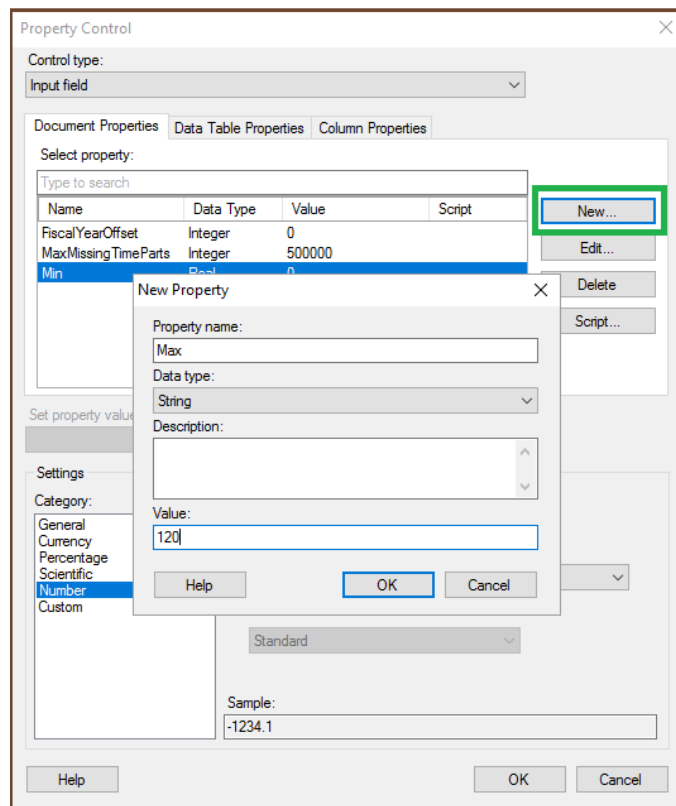


Figure 15: Property control window

Step2: In this step, the x-axis range script is added by first defining the variable names such as “Viz1” and then defining the “Start” and the “End” range values as shown in the figure 16 below.

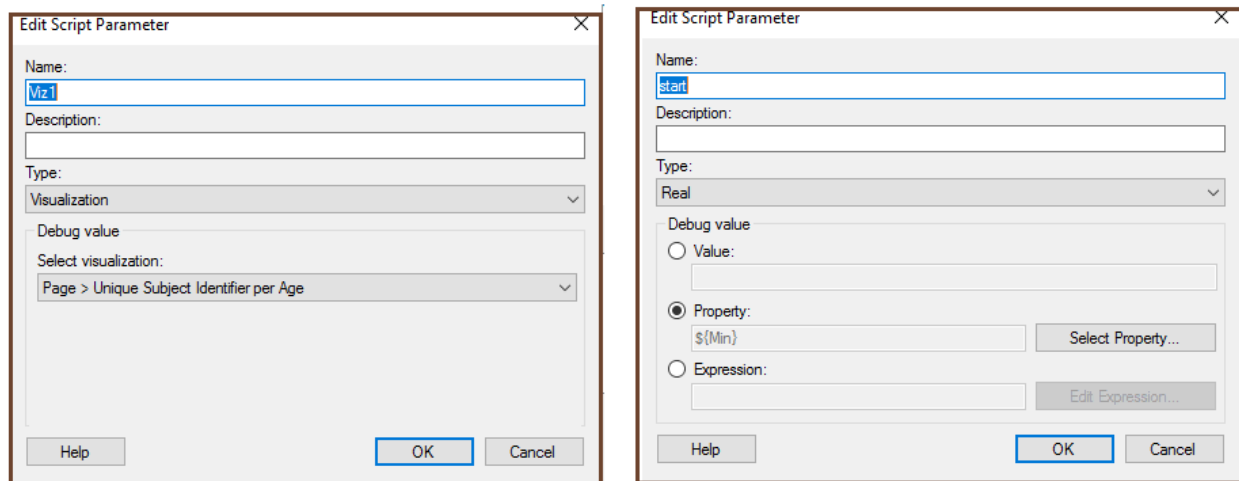


Figure 16: Script Parameter Window

Step4: Figure 17 shows the script to add the dynamic range for x-axis.

Script name: IronPython version:

Description:

Script:

```

1 from Spotfire.Dxp.Application.Visuals import VisualContent, AxisRange
2
3 viz= viz1.As[VisualContent]()
4 viz.XAxis.Range = AxisRange(start,end);
5

```

Script parameters:

Name	Description	Type	Value
Viz1		Visualization	Page > Uni...
start		Real	\${Min}
end		Real	\${Max}

Buttons: Add... Edit... Remove

Run Script Undo

☒ Execute in transaction

Output:

Buttons: Help OK Cancel

Figure 17: Adding new script for setting x-axis range

Example 6

In this example, we are creating a customized function for computing average to draw a line based on the race category using R as shown in figure 18 below. The user could choose a value from the drop-down list in the race category.

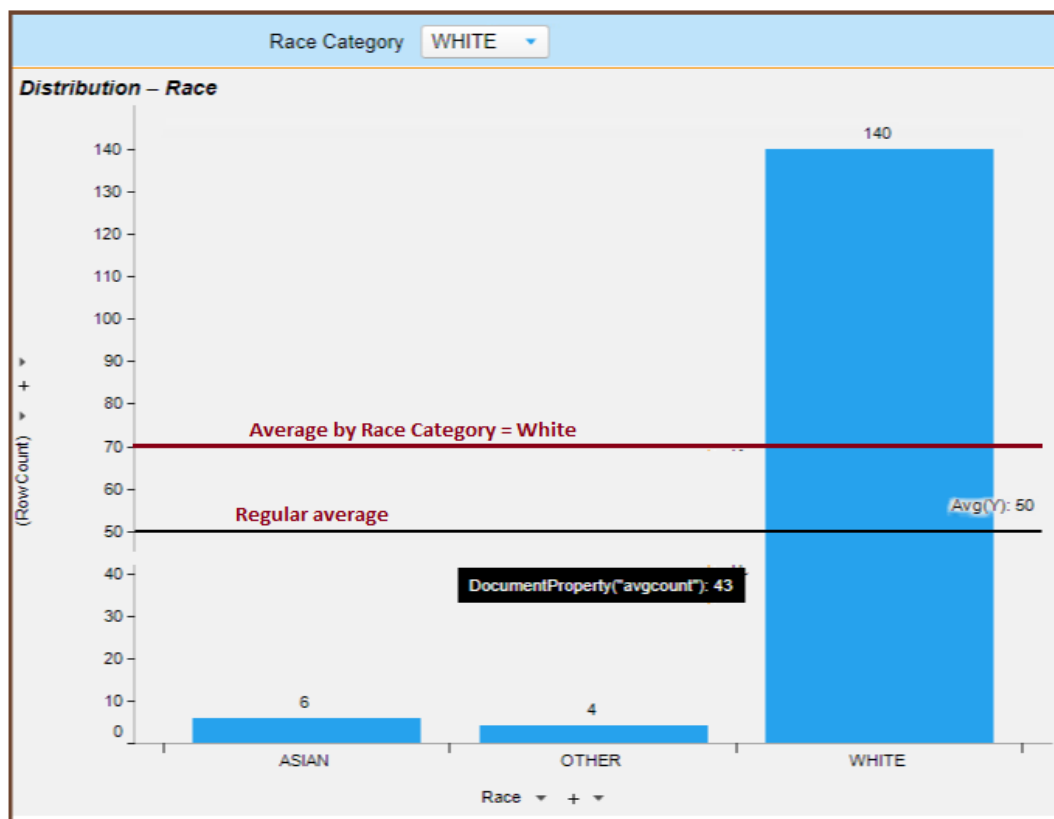


Figure 18: Output panel with dynamic average option

The following steps may be followed to add a customized function using the R script.

Step 1: Insert a text area by selecting the “insert” menu and choosing “text area” option and then add a dropdown field with values from RACE.

Step 2: Create a document property “avgcount” and assign a value 0.

Step 3: Add a data function using menu option Tools→Register Data Function

Step 4: Add below code to scrip area

```
count<-c(Fcnt,Mcnt)
avgcount<-mean(count)
```

Step 5: Define input parameters “Fcnt and Mcnt” as “Single, Real, and Single Real”.

Step 6: Define output parameter “avgcount” as Real.

Step 7: Run the Script from the top menu, which will enable you to enter the custom expression for each parameter.

Step 8: Give the

Name as “CalAvg” (You could give your own name here)

Enable the “Refresh function Automatically” option.

Define input parameter as “Expression and add below script for each input parameter:

```
Fcnt: sum(if([Sex]="F") and ([Race]=Documentproperty("racedropdown")),1,0))
```

```
Mcnt: sum(if([Sex]="M") and ([Race]=Documentproperty("racedropdown")),1,0))
```

Define Output parameter as “Document Property→avgcount”.

Step 9: Click ok and save the new function. You could check the data functions from the menu

Edit→ Data Function Properties

Step 10: Add a line to the graph by following steps.

Choose “Properties→Lines and Curves”

Add a New “Straight Horizontal Line” and choose “Custom Expression” and add below expression

```
DocumentProperty("avgcount")
```

CONCLUSION

Spotfire is rich in graphs and three-dimensional charts that provide the graphical representation of your data. The tool offers an all-inclusive platform for visualization, interaction and sharing of data. The option that allows users to use scripting languages to customize the graphs makes Spotfire highly valuable especially in the clinical trial process. The scripts written in Python, R and other languages allows users to incorporate very beneficial tools into the analysis.

REFERENCES

What is the Filters Panel?, Available at URL:

https://docs.tibco.com/pub/spotfire/7.0.1/doc/html/filter/filter_what_is_the_filters_panel.htm

ACKNOWLEDGMENTS

Sincere thanks to Steve Benjamin, Director Statistical Programming, Biostatistics and Global Contracts and Aman Bahl, Associate Director, Statistical Programming, Clinical Division for their vision, great leadership, persistent support, and encouragement throughout and for their valuable assistance in reviewing this paper.

CONTACT INFORMATION

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

Anish Kuriachen
Principal Statistical Programmer, Clinical Division, Syneos Health
Phone: +1-732-567 5975
E-mail: anish.kuriachen@syneoshealth.com
Web: <http://www.syneoshealth.com>

Hrideep Antony
Principal Statistical Programmer, Clinical Division, Syneos Health
Work Phone: +1- 984 459 4785
Email: hrideep.antony@syneoshealth.com
Web: <http://www.syneoshealth.com>

Brand and product names are trademarks of their respective companies.