



Change Document

Dashboard Designer R-3.0

Contents

1. Dashboard Designer.....	3
1.1. Decision Tree	3
1.2. Histogram	4
1.3. Spark Line	5
1.4. Waterfall	7
1.5. Row Aggregation Fields in Grid Dataset Properties.....	8
1.6. Precision.....	9
1.6.1. Grid Components:.....	9
1.6.2. Chart Components	11
1.7. Data Label Properties	13
1.8. Hide Over Bound Data Label	16
1.9. Increase/Decrease Bar Size	17
1.10. Multiple Series Data in Funnel, Inverted Funnel, and Pyramid Charts	18
1.11. Tick Count in Gauge and Semi Gauge.....	19
1.12. Group Name in the Manage Dashboard Components Window	20
1.13. Header Text Alignment and Color Enhancements in Scorecard and Pivot Grid	20
1.14. Scripting Methods Enhancements	22
1.14.1. sdk.getContextMap	22
1.14.2. sdk.toggleComponents	23
1.14.3. sdk.toggleGroups	25
1.14.4. sdk.applyStyles	27
1.14.5. sdk.applyDatasetToRadioButton.....	28
1.14.6. sdk.hideGroup and sdk.showGroup	30
1.14.7. sdk.getConnectionDetails and sdk.log.....	32
1.14.8. sdk.getDashboardJSON.....	33
1.14.9. sdk.exportDashboardJSON.....	34
1.15. Associate Scorecard and Pivot Components with Legends	35
1.16. Heatmap chart: Color Enhancements.....	37
1.17. Chart Title Enhancement	38
1.18. Enhancements in the Dataset Point Shape Properties	39

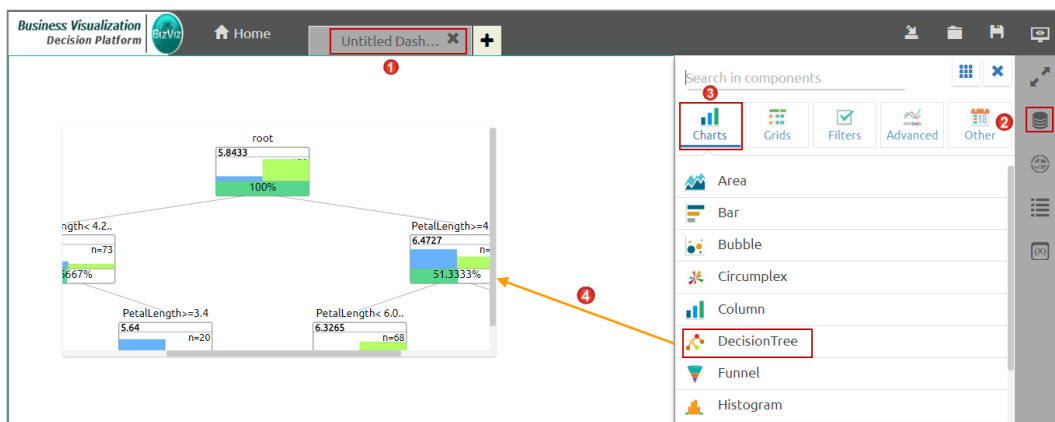
1. Dashboard Designer

1.1. Decision Tree

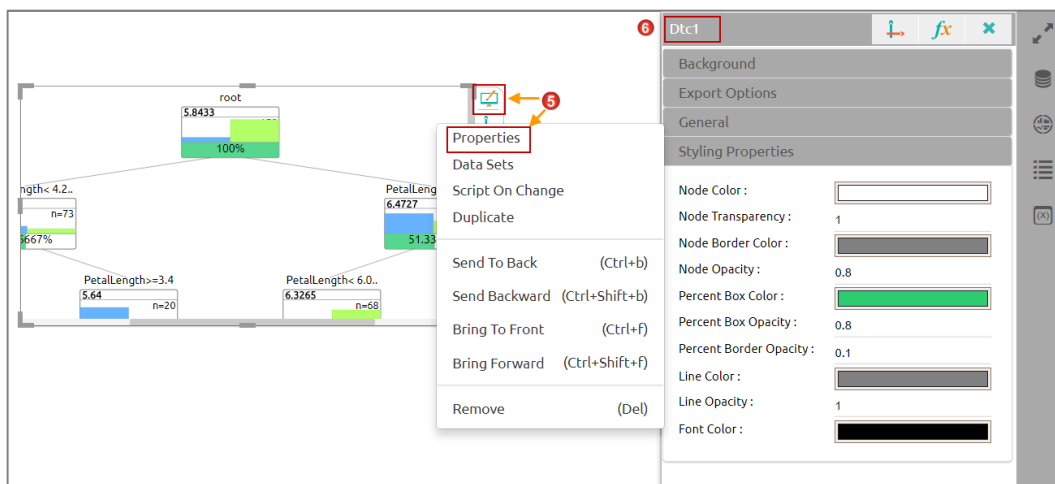
The Decision Tree presents outcomes from a series of decisions via a flow-chart like diagram. Users can easily understand the sequence presented through this graph, so frequently it has been used as a decision-making tool, for analysis, or planning strategy.


The chart can either be used in predictive or descriptive manner. The best situation to use this chart is to study the current market in response to an existing or new product, or to understand the sequential elements involved in the financial matter (loan approval etc.)

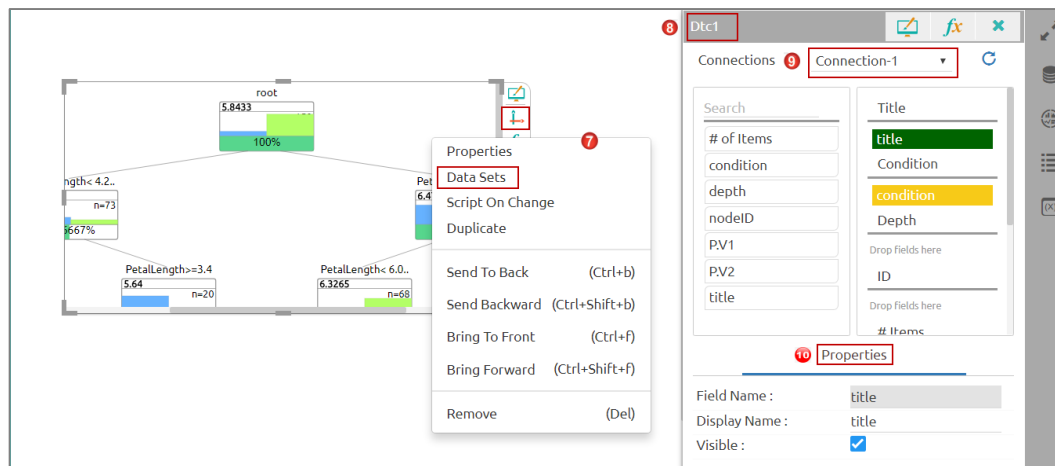
- Navigate to the Designer page.
- Click the 'Component Library' icon.
- Click 'Charts' option from the list of components.
- Select and drag the Decision Tree chart component on the canvas.



- Click the Decision Tree component on the designer canvas.
 - Click the 'Properties' icon to display the chart specific properties.
- OR
- Right-click on the Decision Tree Chart component.
 - Click the 'Properties' option to display the chart specific properties.
 - The chart specific properties will be displayed on the right side of the chart component.



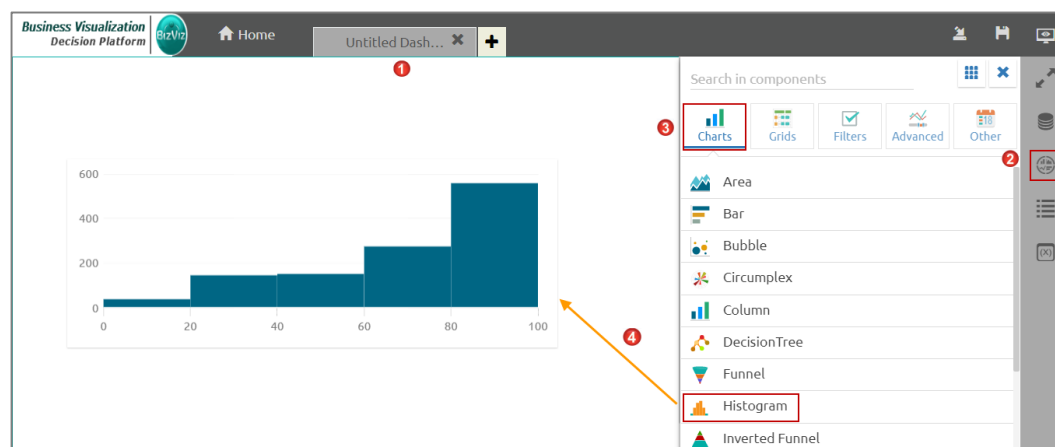
- Click the chart component on the designer canvas.
- Click the 'Dataset' icon 
OR
- Select the 'Data Sets' option by using a right click on the chart component.
- Users will be redirected on the Dataset palette.
- Select a valid connection from the data set palette.
- The chart specific Dataset Properties will be displayed.




1.2. Histogram

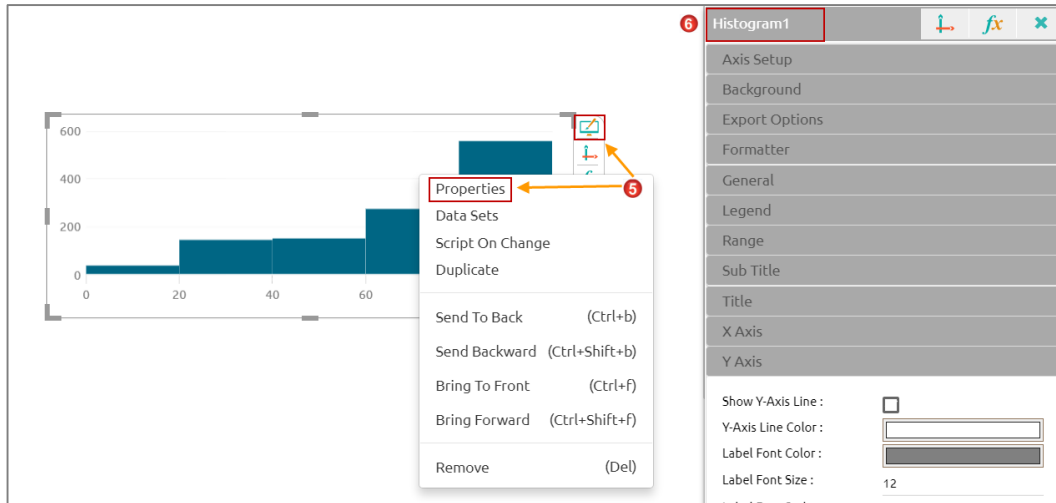
The X-Axis remains a continuous scale in the Histogram chart that makes it different from the Vertical Bar Chart. The Histograms make it easy to grasp where most values fall in a measurement scale, and how much variation is there. It is used to summarize large amount of data or compare process results with specification limits.

- Navigate to the Designer page.
- Click the 'Component Library' icon.
- Click 'Charts' option from the list of components.
- Select and drag the Histogram chart component on the canvas.



- Click the Histogram component on the designer canvas.
- Click the 'Properties' icon  to display the chart specific properties.

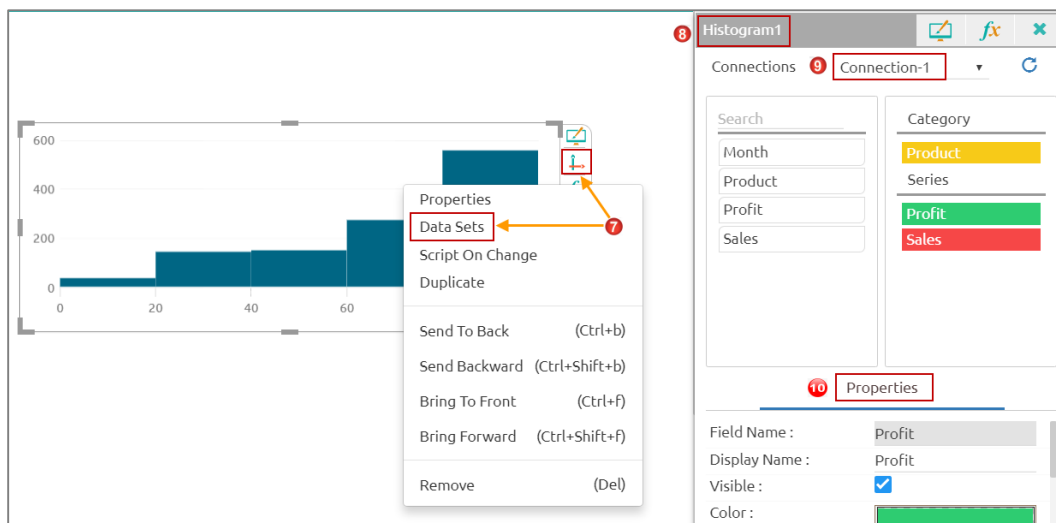
- OR
- Right-click on the Histogram Chart component.
- Click the 'Properties' option to display the chart specific properties.
- The chart specific properties will be displayed on the right side of the chart component.



- Click the chart component on the designer canvas.

- Click the 'Dataset' icon
- OR

- Select the 'Data Sets' option by using a right click on the chart component.
- Users will be redirected on the Dataset palette.
- Select a valid connection from the data set palette
- The chart specific Dataset Properties will be displayed.

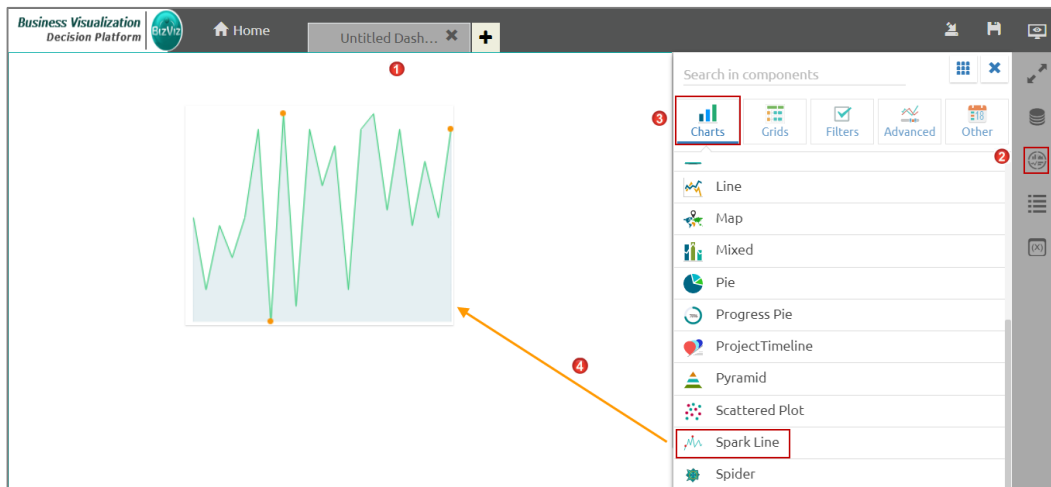



1.3. Spark Line

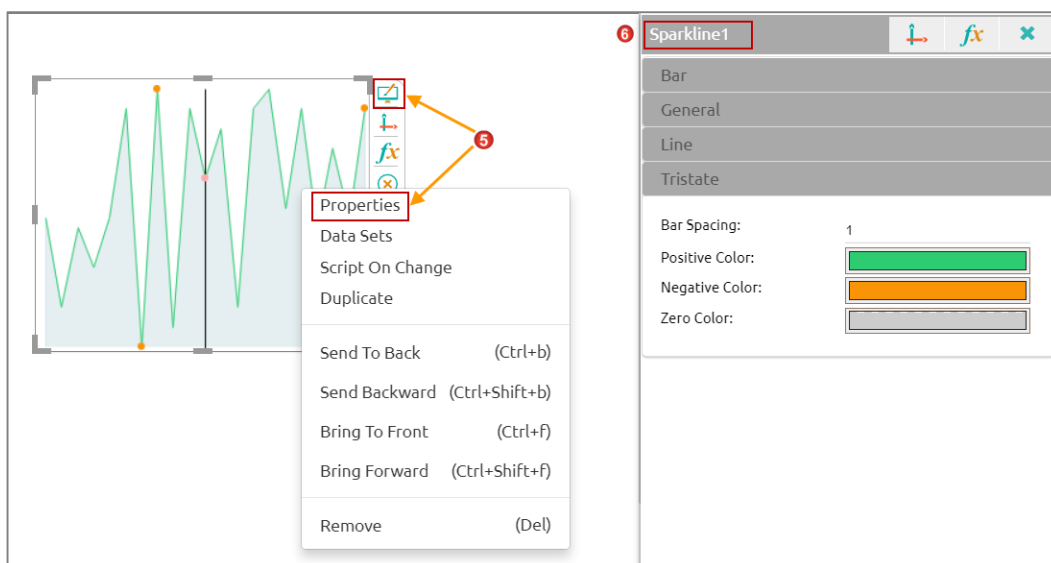
A Sparkline is a very small chart that is drawn without axes and lacks some chart-specific elements (such as legend, title etc.). The Sparkline chart can be easily embedded in the text as it performs more like an inline


element (rendered inside a span) as opposed to the standard charts, which behave like block elements. The main purpose attached with the Sparkline Chart is to show trend of something unique.

- Navigate to the Designer page.
- Click the **'Component Library'** icon.
- Click **'Charts'** option from the list of components.
- Select and drag the Decision Tree Chart component on the canvas.



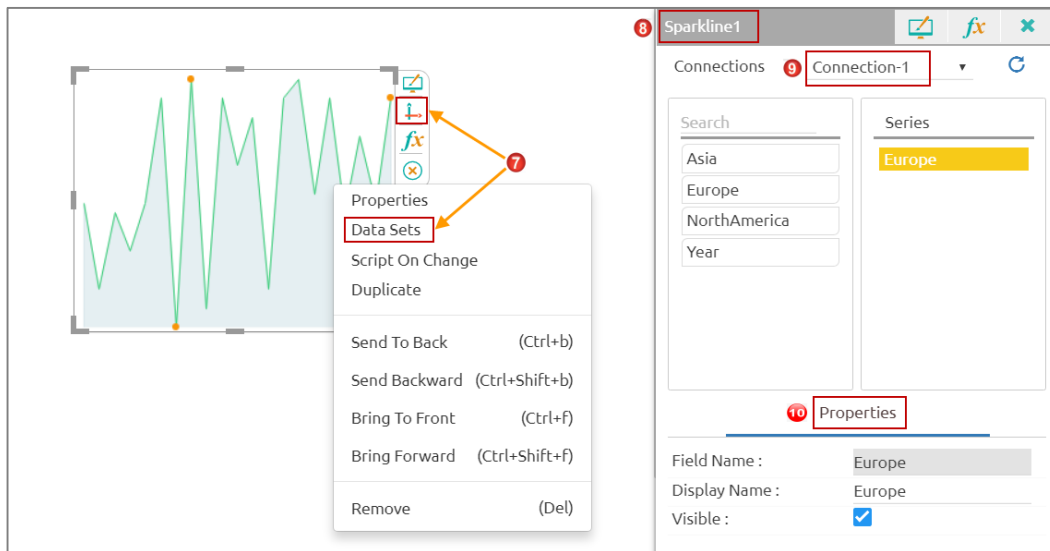
- Click the Spark Line component on the designer canvas.
 - Click the **'Properties'** icon  to display the chart specific properties.
- OR
- Right-click on the Decision Tree Chart component.
 - Click the **'Properties'** option to display the chart specific properties.
 - The chart specific properties will be displayed on the right side of the chart component.



- Click the chart component on the designer canvas.
- Click the **'Dataset'** icon 

OR

- Select the 'Data Sets' option by using a right click on the chart component.
- Users will be redirected on the Dataset palette.
- Select a valid connection from the data set palette.
- The chart specific DataSet Properties will be displayed.

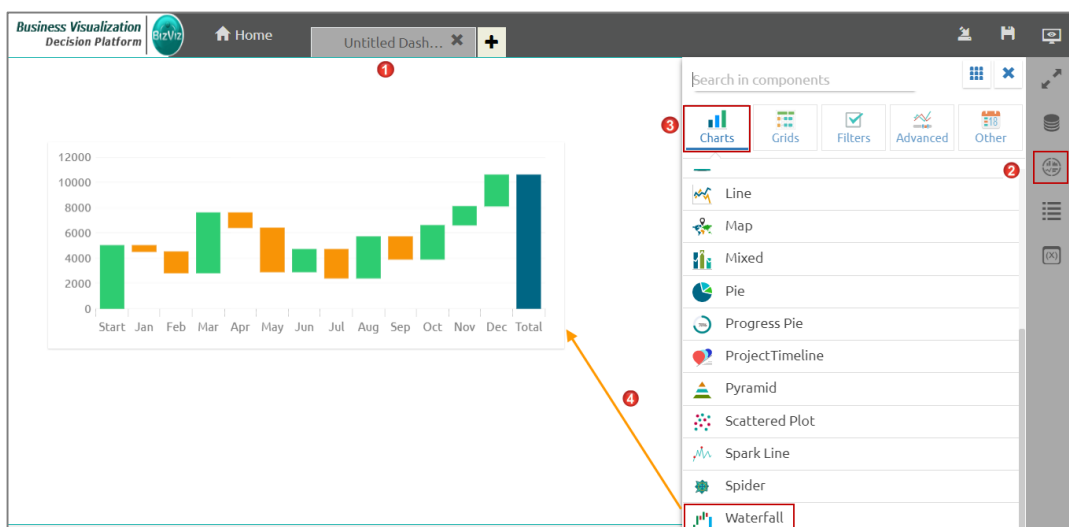


1.4. Waterfall

A Waterfall Chart is a form of data visualization that helps in understanding the cumulative effect of sequentially introduced positive or negative values. The waterfall chart is also known as a flying bricks chart or Mario chart due to the apparent suspension of columns (bricks) in mid-air.

The Waterfall Chart is used to show how an initial value is increased and decreased by a series of intermediate values, leading to a final value.

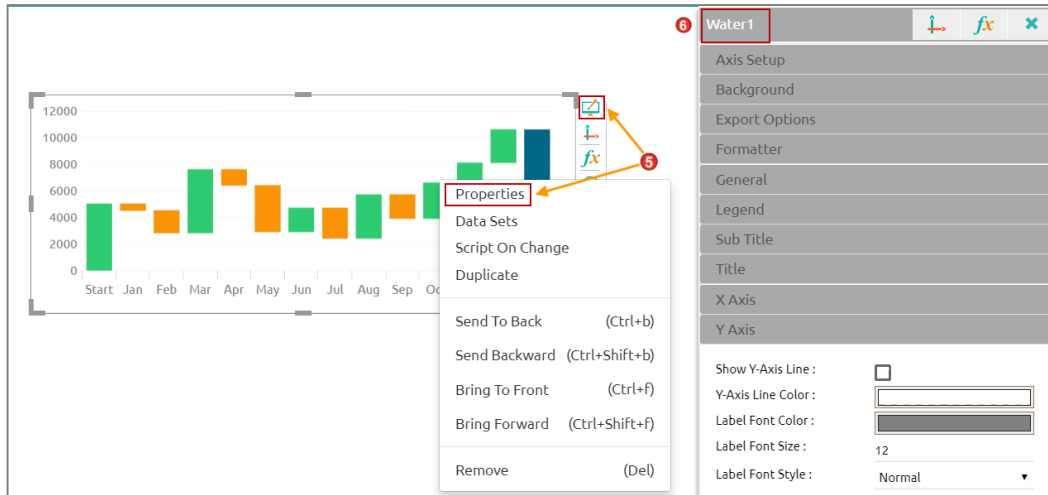
- Navigate to the Designer page.
- Click the 'Component Library' icon.
- Click 'Charts' option from the list of components.
- Select and drag the Waterfall Chart component on the canvas.



- Click the Waterfall component on the designer canvas.
- Click the 'Properties' icon to display the chart specific properties.

OR

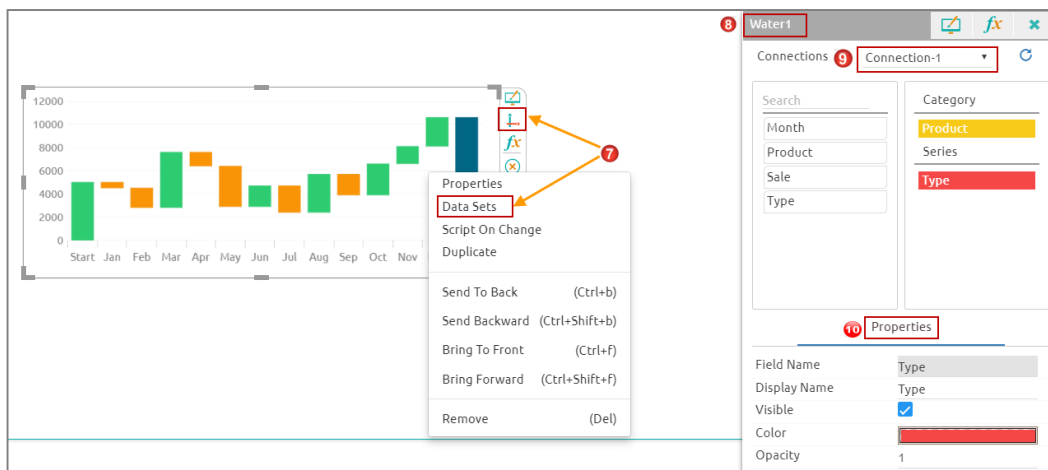
- Right-click on the Waterfall Chart component.
- Click the 'Properties' option to display the chart specific properties.
- The chart specific properties will be displayed on the right side of the chart component.



- Click the chart component on the designer canvas.
- Click the 'Dataset' icon

OR

- Select the 'Data Sets' option by using a right click on the chart component.
- Users will be redirected on the Dataset palette.
- Select a valid connection from the data set palette.
- The chart specific DataSet Properties will be displayed.

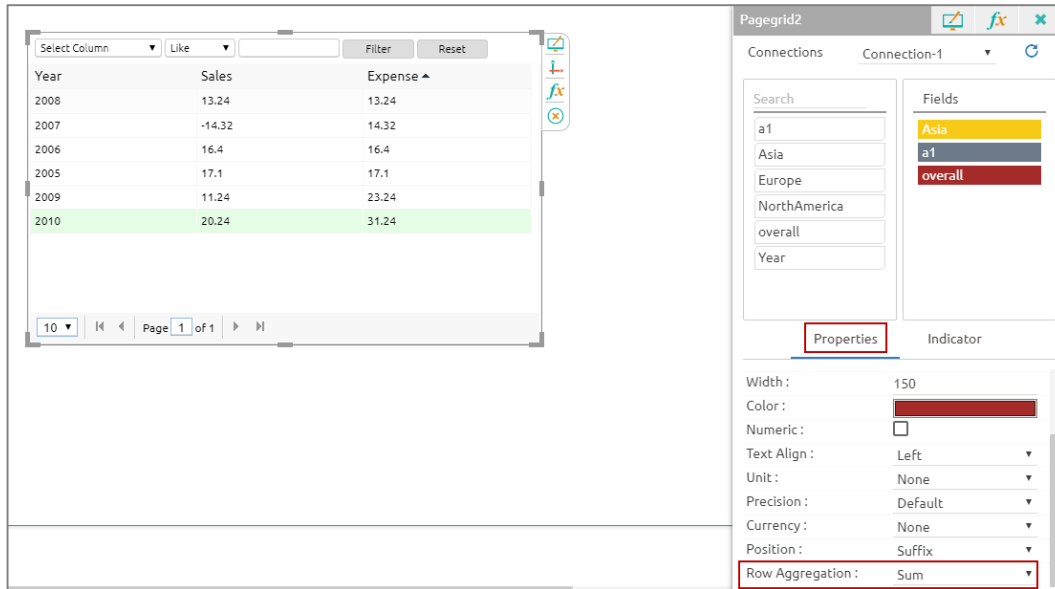


1.5. Row Aggregation Fields in Grid Dataset Properties

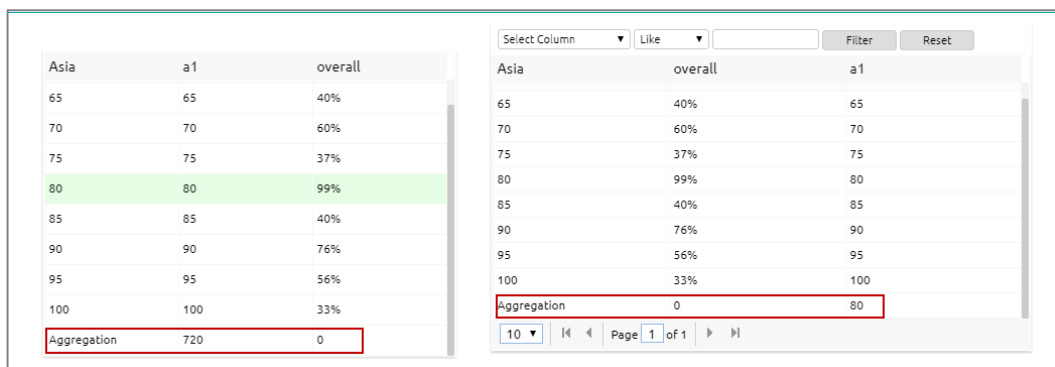
The Row Aggregation field has been provided as a drop-down menu in the dataset properties of the Grid components. The row aggregation drop-down contains None, Sum, Average, Count, Minimum, Maximum options. The default row aggregation option is 'None'. field will provide various aggregation options via a drop-down menu.

- Select a Grid component and drag it on the designer canvas.
- Connect the grid component with a valid data connection and open the dataset properties.

- Select an aggregation option from the Row Aggregation drop-down menu.



- Click 'Preview'.
- The Row Aggregation will appear as the last row in the preview.
- The following image displays preview of Simple Grid and Paging Grid components:
 - The selected row aggregation type is 'Sum' for the Simple Grid component.
 - The selected row aggregation type is 'Average' for the Paging Grid component.



Note: The default row aggregation option is 'None'.

1.6. Precision

Users can change/control the precision option in Grid and Chart components.

1.6.1. Grid Components:

- Select and drag a Grid component on the designer canvas.
- Connect with a valid data connection.

Year	Sales	Expense
2010	20.24	31.24
2009	11.24	23.24
2005	17.1	17.1
2006	16.4	16.4
2007	-14.32	14.32
2008	13.24	13.24

- Select field specific precision option using the dataset properties.

Properties

Display Name : a1

Visible :

Frozen Column :

Width : 150

Color : [Color Picker]

Numeric :

Text Align : Left

Unit : None

Precision : .00000000

Currency : None

Position : Suffix

Row Aggregation : None

- Click 'Preview'.
- The selected precision option can be displayed for the selected field in the dataset.

NorthAmerica	a1
50	60.00000000
60	65.00000000
100	70.00000000
150	75.00000000
170	80.00000000
190	85.00000000
210	90.00000000
225	95.00000000

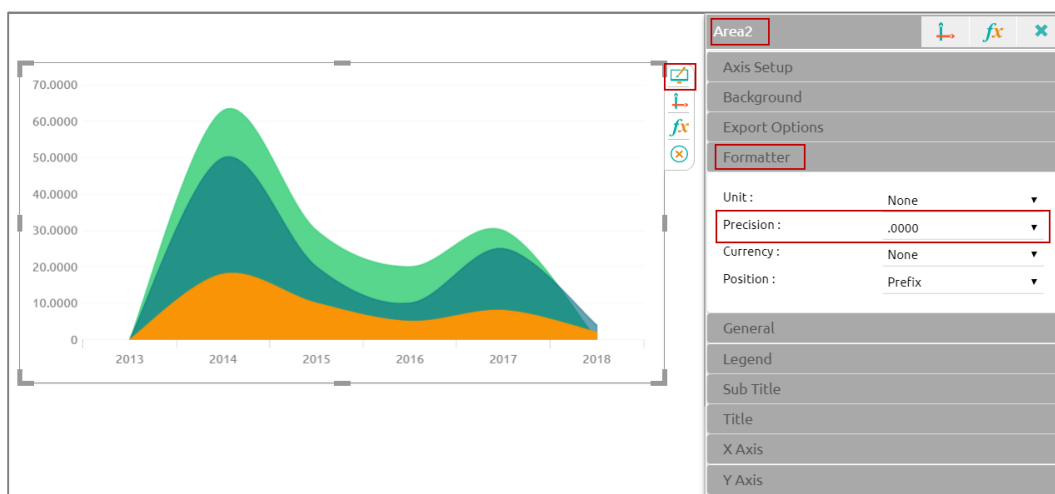
Note: Users can select field specific precision option for each field in a grid.

CalcField1	CalcField2	CalcField3	CalcField4	CalcField5	CalcField6	CalcField7	CalcField8
0.11494252873563218	0	0.1	0.11	0.115	0.1149	0.114943	0.11494253
0.22988505747126436	0	0.2	0.23	0.230	0.2299	0.229885	0.22988506
0.3448275862068966	0	0.3	0.34	0.345	0.3448	0.344828	0.34482759
0.45977011494252873	0	0.5	0.46	0.460	0.4598	0.459770	0.45977011
0.5747126436781609	1	0.6	0.57	0.575	0.5747	0.574713	0.57471264
0.6896551724137931	1	0.7	0.69	0.690	0.6897	0.689655	0.68965517
0.8045977011494253	1	0.8	0.80	0.805	0.8046	0.804598	0.80459770
0.9195402298850575	1	0.9	0.92	0.920	0.9195	0.919540	0.91954023
1.0344827586206897	1	1.0	1.03	1.034	1.0345	1.034483	1.03448276
0.7471264367816092	1	0.7	0.75	0.747	0.7471	0.747126	0.74712644

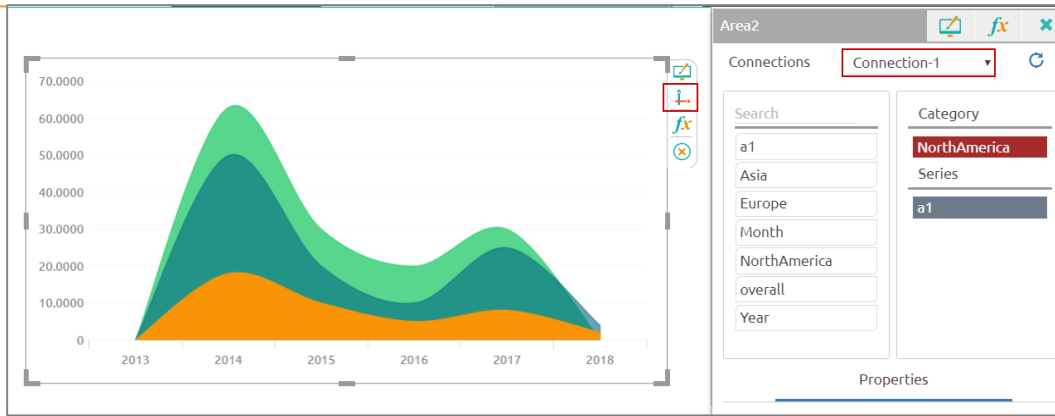
1.6.2. Chart Components

Precision options can be selected from the Formatter properties of a chart component containing Y Axis.

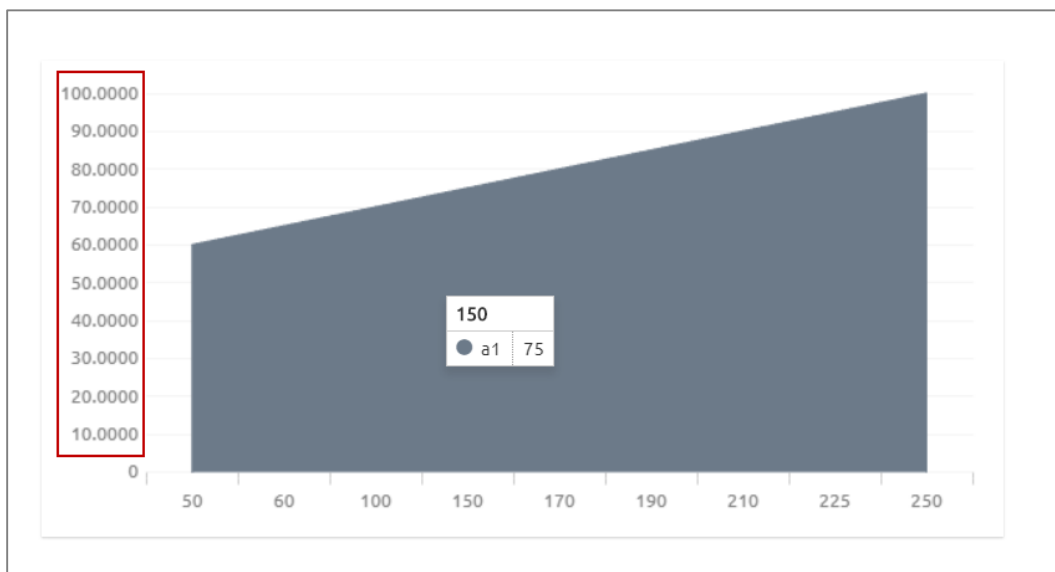
- Select and drag a chart component (containing Y-Axis) on the designer canvas.
- Select a Precision option from the Formatter properties of the chart.



- Connect the chart with a valid data connection.



- Click 'Preview'.
- The selected precision value will be applied to the Y-Axis data.



Note: Default Precision value will display the actual decimal value as given in the dataset. Users need to change the script to change default precision value for a chart.

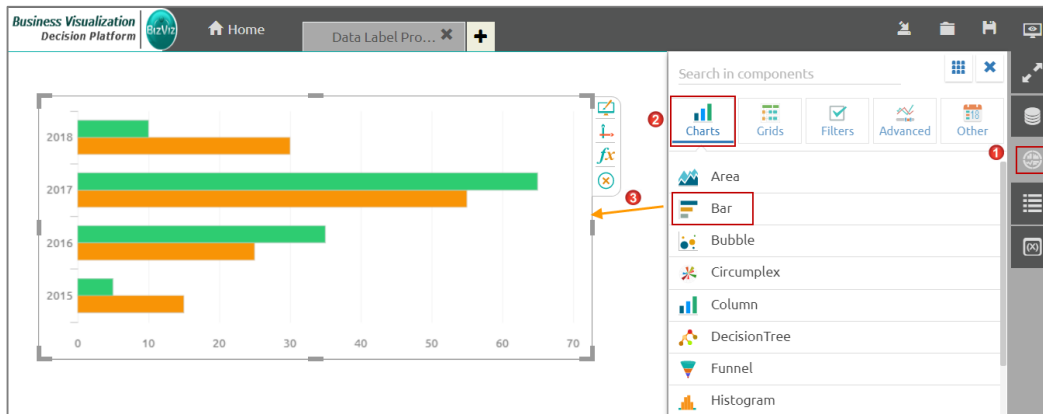
```

var ch = sdk.getWidget("area1");
ch.m_tooltipprecision = "default";
var ch = sdk.getWidget("area2");
ch.m_tooltipprecision = "2";
var ch = sdk.getWidget("area3");
    
```

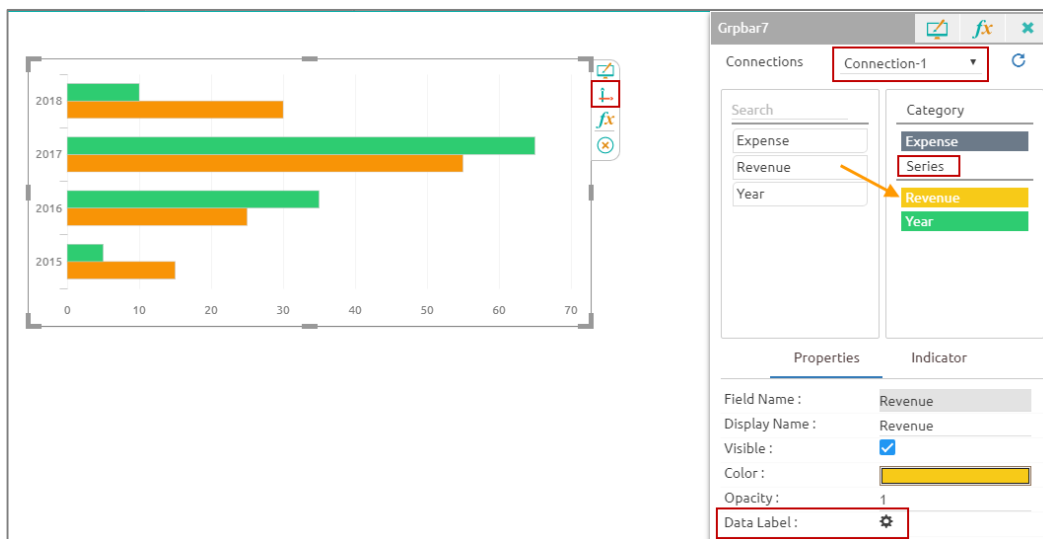
1.7. Data Label Properties

Users can insert data labels in the selected charts by configuring the data label properties. The Data Label properties option is provided under the 'Series' properties.

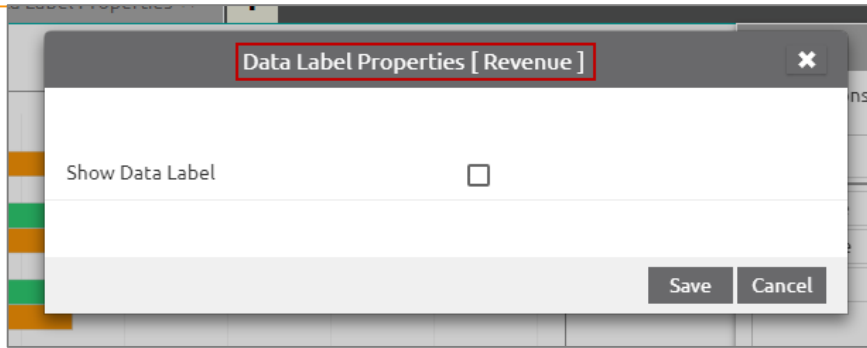
- Click the 'Component Library' icon from the Designer Page.
- Select and drag a chart component on the designer canvas (In this case, out of Area/Time/Line/Bar/Group Bar/Column charts)



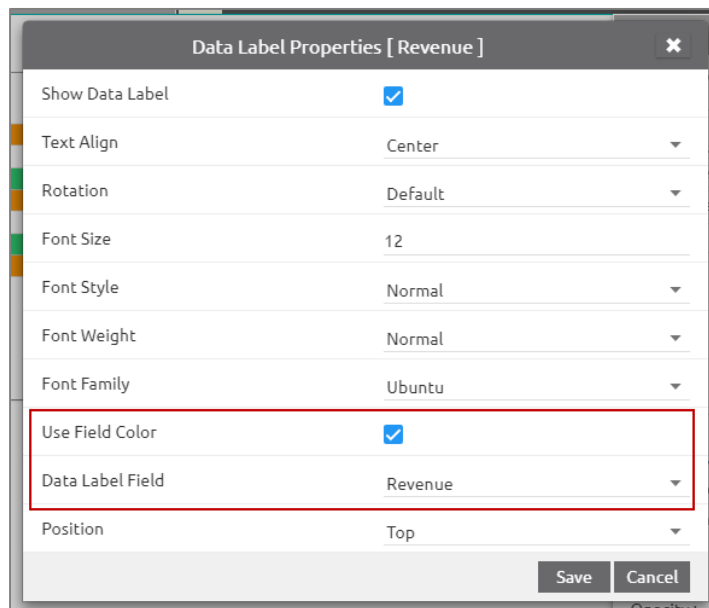
- Click the 'Dataset' option
- Connect the chart with a valid data connection.
- Select a field as Series.
- The 'Data Label' field will be displayed below as a Series properties.



- Click the 'Data Label' option from the Series Properties.
- A new window will pop-up.
- Select 'Show Data Label' option by check marking the box.

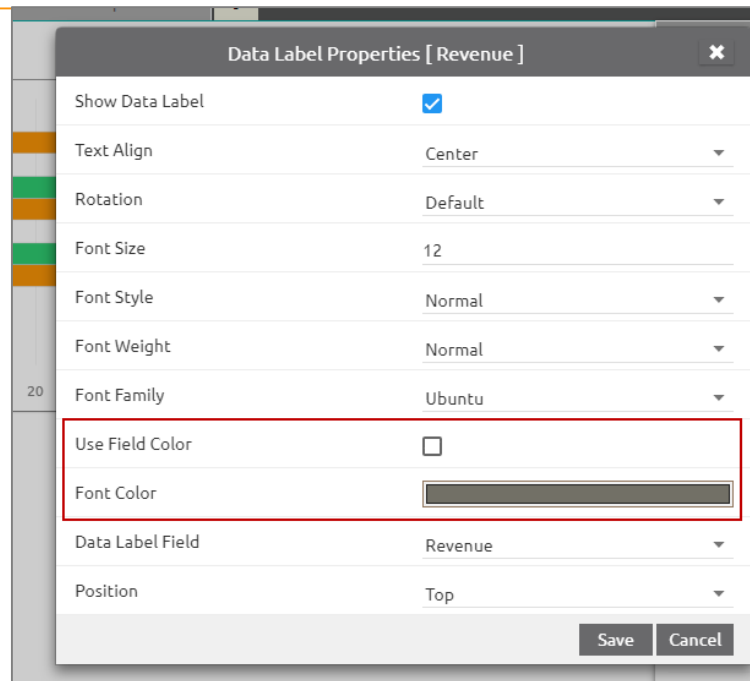


- Users will be directed to configure the Data Label properties for the selected series.
- Configure the following information:
 - Show Data Label: Enable this option to show the data label.
 - Text Align: Select a text alignment option from the drop-down menu.
 - Rotation: Select a rotation option from the drop-down menu.
 - Font Size: Set font size.
 - Font Style: Select a font style from the drop-down menu.
 - Font Weight: Select font weight option from the drop-down menu.
 - Font Family: Select a font type from the drop-down menu.
 - Use Field Color: Enable/disable the option by check marking in the box.
 - **Data Label Field:** This field appears, if 'Use Field Color' option has been enabled.

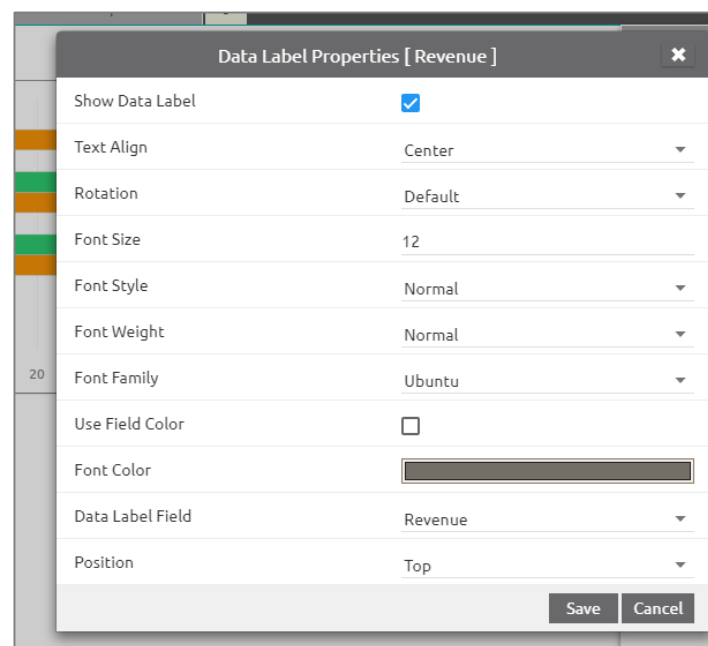


OR

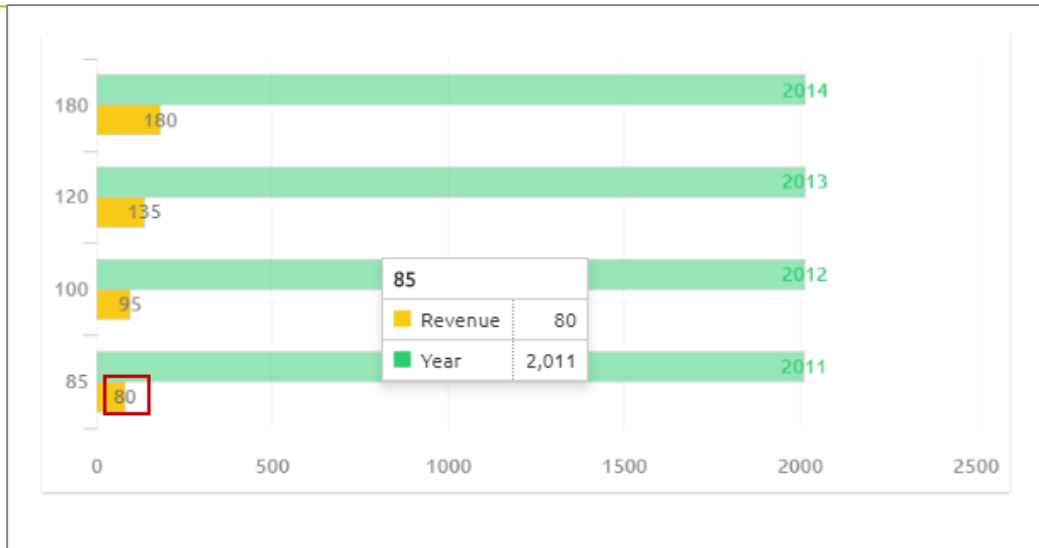
- **Font Color:** Users need to select a font color from the menu if the 'Data Label Field' option has not been selected.



- Position: Select the label position from the drop-down menu.
- Click 'Save'.

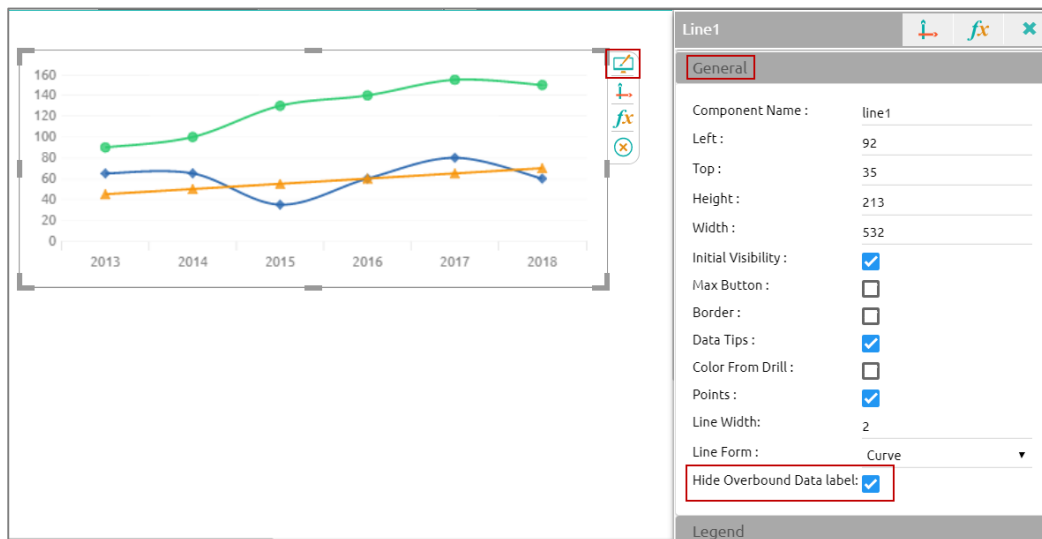


- Click 'Preview' option 
- Users will be redirected to the preview screen displaying the chart with data Labels.

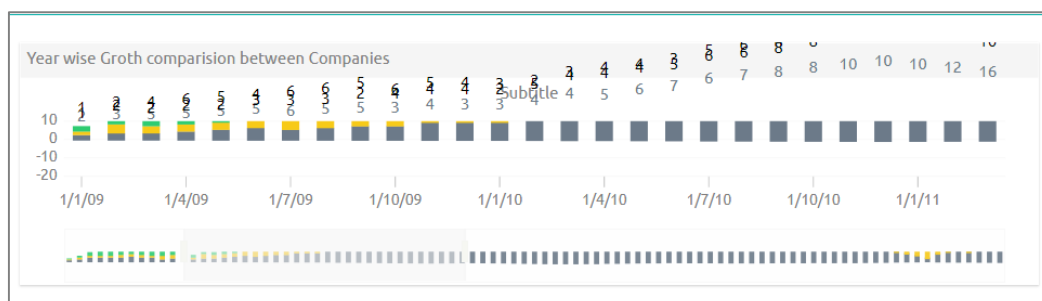



1.8. Hide Over Bound Data Label

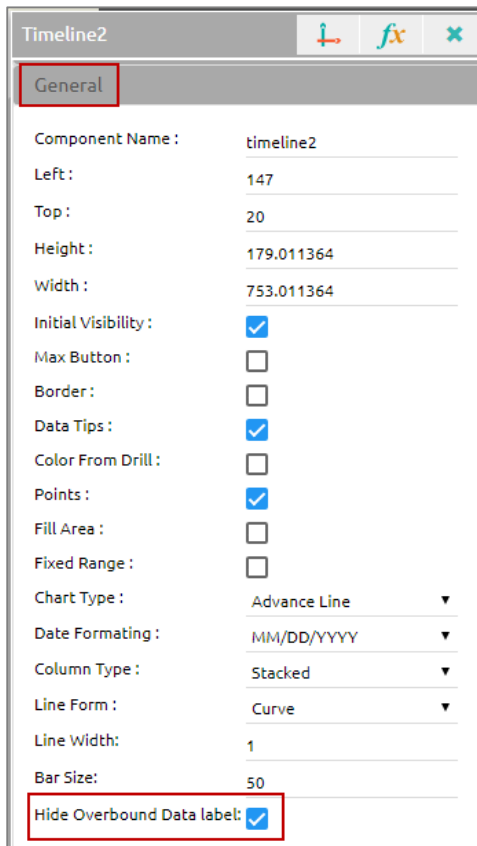
- Select chart component and drag it on the designer canvas (In this case, Timeseries chart component has been selected).



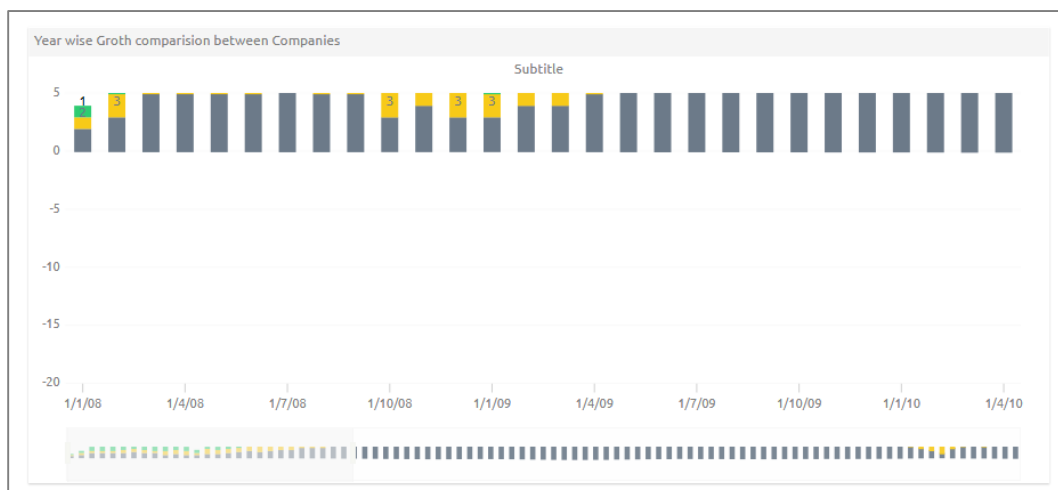
- Connect the chart component with a valid data connection and open the dataset properties.
- Configure Data Label properties for the required fields.
- Click 'Preview'.



- Click 'Properties' icon  on the chart.
- Select 'General' option from the displayed properties menu.
- Enable 'Hide Overbound Data Label' by check marking in the box.



- The overbound chart data labels will be hidden in the preview.



1.9. Increase/Decrease Bar Size

Users can increase or decrease bar size in the Bar, Column, Group Bar, Mixed, Timeline, and Waterfall charts via the properties palette.

- Select and Drag a chart component onto the designer canvas (In this case, Mixed Chart has been selected).

- Click the 'Properties' icon.
- Open 'General' Properties.
- Use 'Bar Size' field to increase/decrease the column bar size (Default bar size is 65).



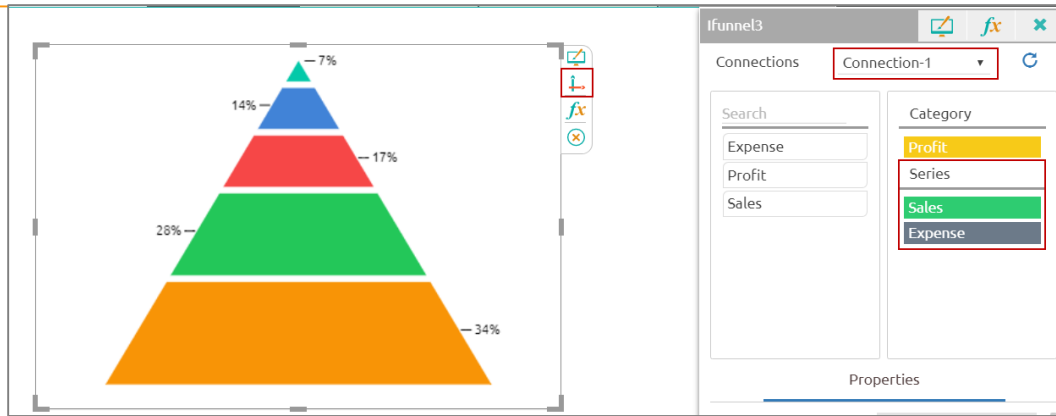
- Various Bar Sizes in the Mixed Chart components.



1.10. Multiple Series Data in Funnel, Inverted Funnel, and Pyramid Charts

Users can select multiple categories as series for the Funnel, Inverted Funnel, and Pyramid charts.

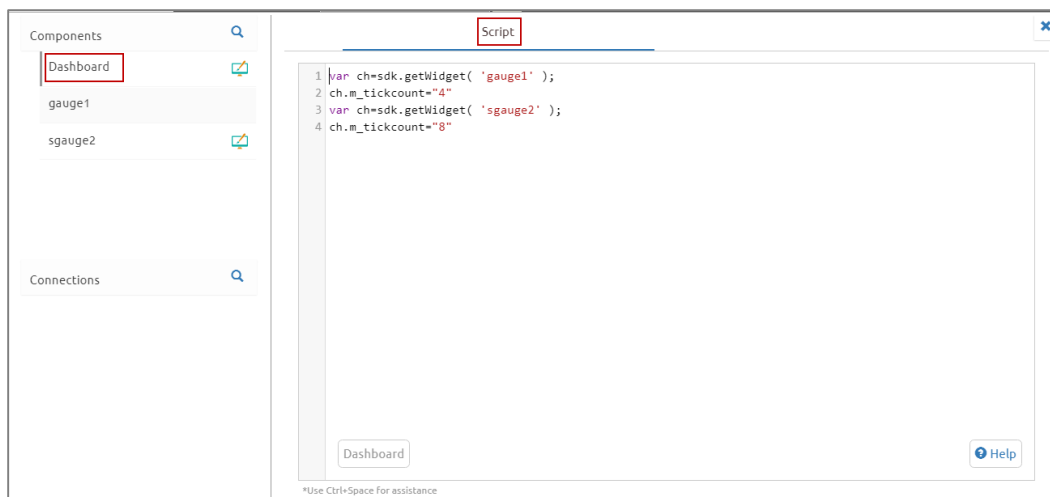
- Select a chart component and drag it on the designer canvas. (In this case, the selected chart component is Funnel)
- Connect the chart component with a valid data set.
- Select multiple fields as Series.



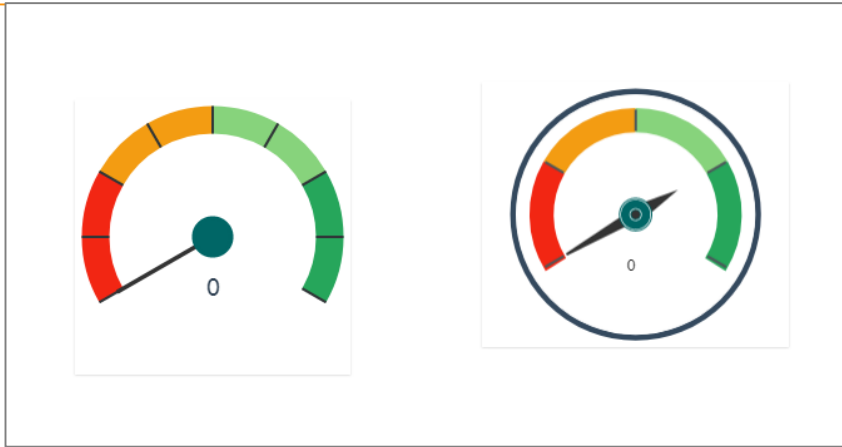
1.11. Tick Count in Gauge and Semi Gauge

Tick count can be changed for the Gauge and Semi Gauge components by changing the dashboard script using the script window.

- Select a Gauge/Semi-Gauge component and drag it on the designer canvas.
- Click 'Script on Change' icon
- Define the tick counts in the gauge/semi gauge components using the script written for Dashboard.



- Click 'Preview'.
- The tick count will be changed according to the script.



1.12. Group Name in the Manage Dashboard Components Window

The component group name can be displayed in the Manage Dashboard Component window.

- Click **'Manage Dashboard Components'** icon on the Dashboard Designer.
- A new window will be opened on the right side of the designer canvas.
- Click the **'Add Group'** option.
- The created group will be displayed in the Manage Dashboard Components window.



1.13. Header Text Alignment and Color Enhancements in Scorecard and Pivot Grid

Users can provide column specific alignment for text and color in the Scorecard and Pivot Grid using the Dataset properties.

- Select and drag the scorecard/pivot grid component on the designer canvas.
- Connect the dragged component with a valid dataset.
- Enable **'Use Field Color'** option using the **'Header Style'** properties.

City	Population	Revenue	Indicator
USA			
California			
Los Angeles	4500	75	↑
San Francisco	2500	45	↑
San Jose	1000	10	↑
San Diego	500	5	↓
Santa Clara	300	3	↓
New York			
NY City	6000	78	↑
Rochester	278	23	↑
Buffalo	100	2	↓
Texas			

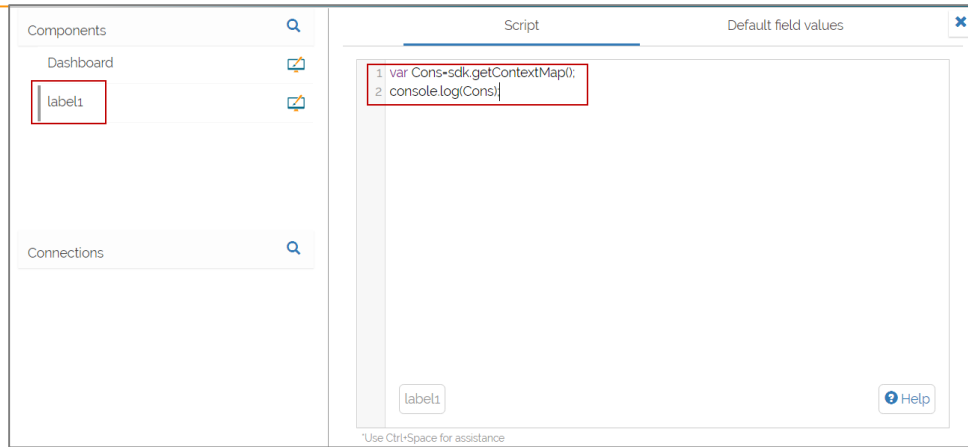
- Connect the component with a valid data set and configure the required properties information (If required).

City	Population	Revenue	Indicator
USA			
California			
Santa Clara	300	3	↓
San Diego	500	5	↓
San Jose	1000	10	↑
San Francisco	2500	45	↑
Los Angeles	4500	75	↑
New York			
Buffalo	100	2	↓
Rochester	278	23	↑
NY City	6000	78	↑
Texas			
San Antonio	1500	33	↑
Dallas	2500	34	↑

- Click 'Preview'.
- The column colors will be displayed as background color for the column headers.

ShipCountry	ShipState	ShipCity
USA	NY	New York
USA	NY	New York
USA	NY	New York
USA	NY	New York
USA	NY	New York
USA	NY	New York
USA	NY	New York
USA	NY	New York
USA	NY	New York
USA	NY	New York
USA	NY	New York
USA	FL	Las Vegas
USA	FL	Las Vegas
USA	IL	Las Vegas

The selected Text alignment is 'left' for the below-displayed chart:



- Click 'Preview'.
- Click the label component via the preview screen.
- Open the Developer's Tools to see the contextual details of the logged in user (Click 'F12' Shortcut-key to open the Developer's Tools).



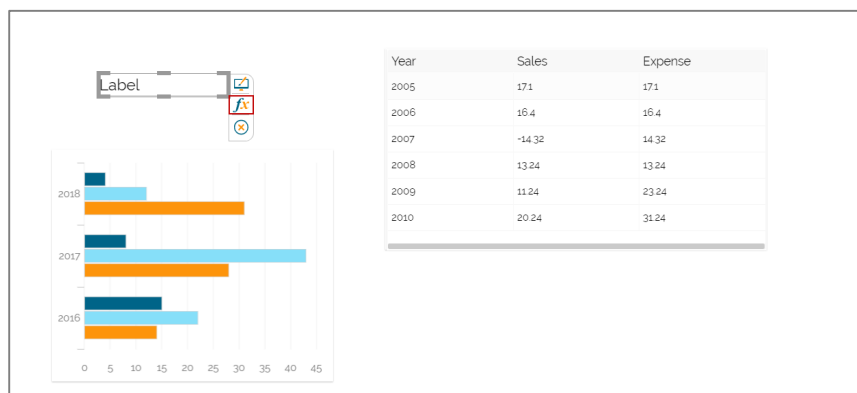
1.14.2.

sdk.toggleComponents

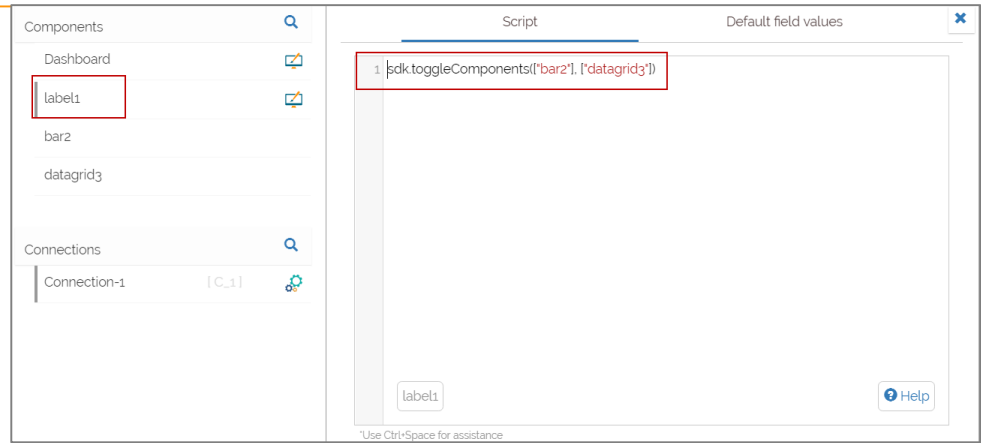
This method helps to toggle the visibility of the selected components.

The script is described through the following example:

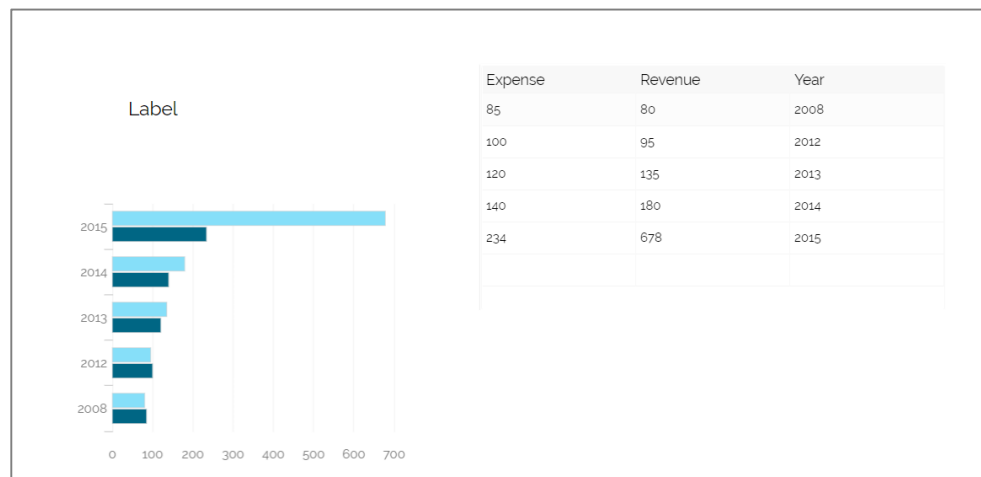
- Drag and drop the required components onto the canvas (In this case, the used components are a 'Label', a 'Bar chart' and a 'Data Grid').



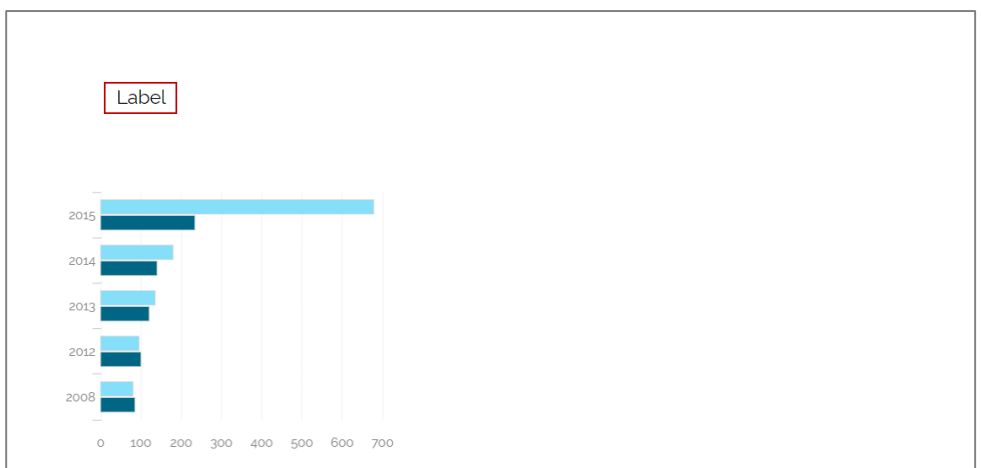
- Open the component scripting window of the 'Label' component by clicking the 'Script on Change' option
- Write the script as displayed in the following image:



- Click 'Preview'.
- Users will be redirected to see the preview of the dashboard (displaying both the components).



- Users will be displayed the preview of either of the components (Either Bar or Data Grid) in turns while clicking on the 'Label' component.



Expense	Revenue	Year
85	80	2008
100	95	2012
120	135	2013
140	180	2014
234	678	2015

1.14.3. sdk.toggleGroups

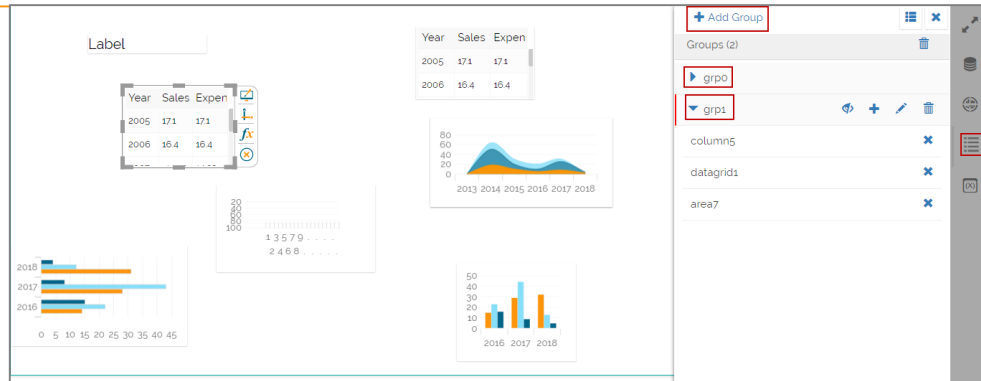
Users can toggle preview of the selected groups in turns by using this script.


The script is described through the following example:

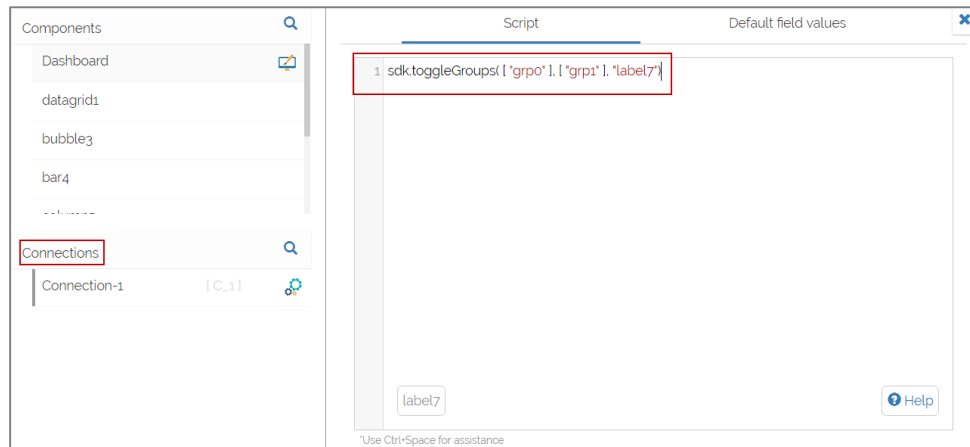
- Drag and drop the required components on the canvas to form groups (In this case the selected components are: one 'Label', two 'Data Grids', and Charts-'Area', 'Bar', 'Bubble', and 'Column').



- Click the 'Manage Dashboard Components' option
- Users will be redirected to create groups and add the required components into a group.
 - In this case, the created groups are 'grp0' and 'grp1'.
 - Components added to the groups are as listed below:
 - grp0: bar4, bubble3, datagrid7
 - grp1: column5, datagrid1, area7



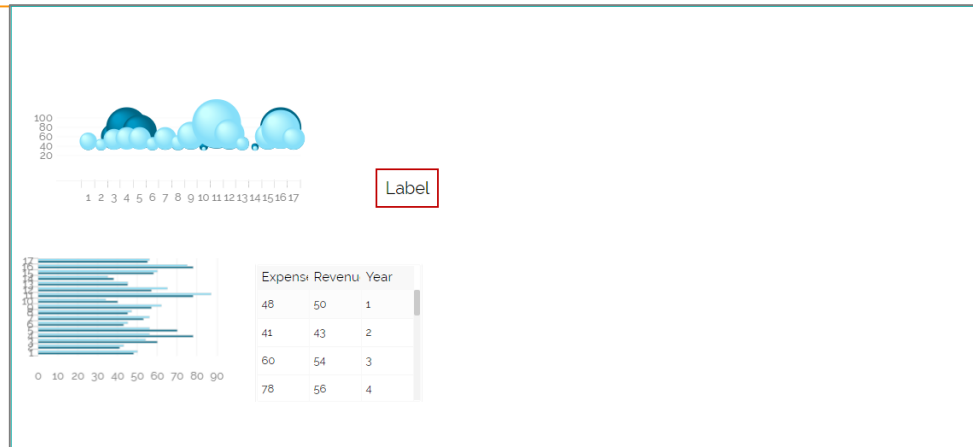
- Click the 'Script on Change' icon  to open the Scripting Window (In this case the Label component is used to open the scripting window and the same component will be considered as the toggle key).
- Write a script under the 'Connections' section as displayed in the following image:



- Click 'Preview'.
- Users will be redirected to see the preview of the dashboard.




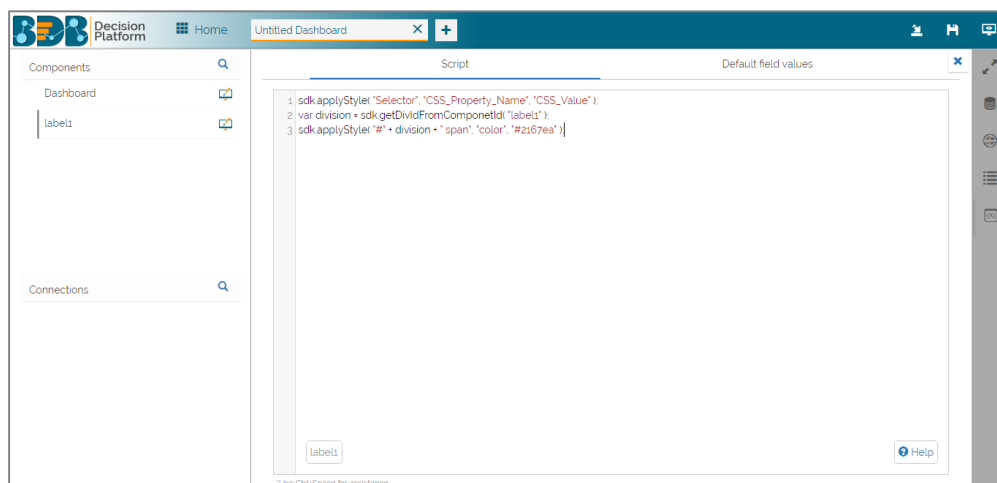
- The group preview will toggle in turns between the two groups by clicking the 'Label' component as displayed in the following images:



1.14.4. sdk.applyStyles

Users can apply a complete object of CSS styles to the selector by using this script. The script is described through the following example:

- Drag and drop a component onto the canvas (In this case the used component is 'Label').
- Click the 'Script on Change' icon  of the 'Label' component to open the Scripting Window.
- Write the script as mentioned in the following image:



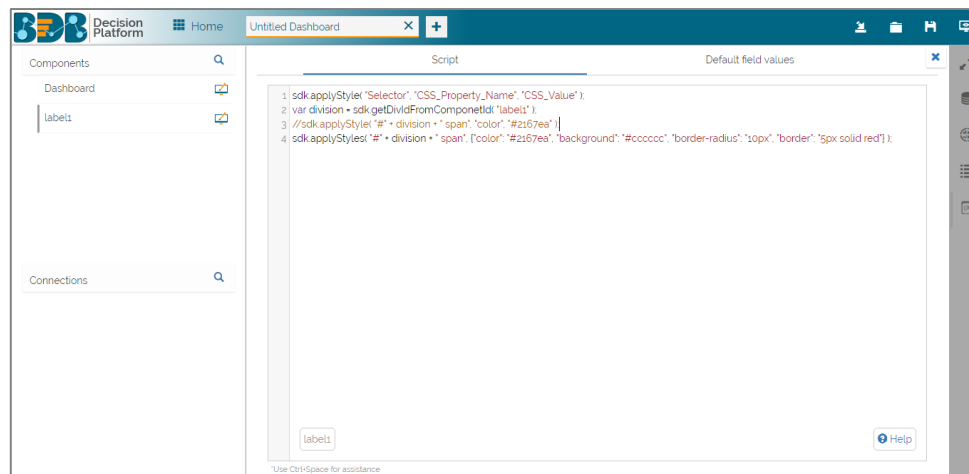
- Click 'Preview'.
- Click the 'Label' component.

- The selected style will be applied to the label component.

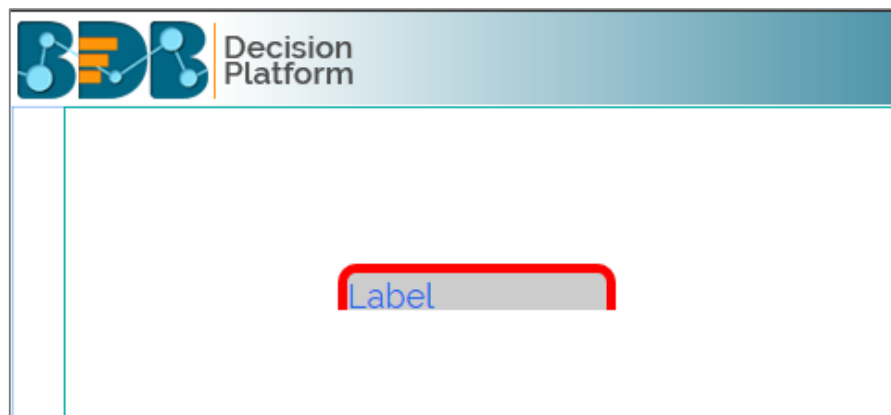


Note: By using the same process different styles can be applied to the component objects.

- Write the script as displayed in the following image:




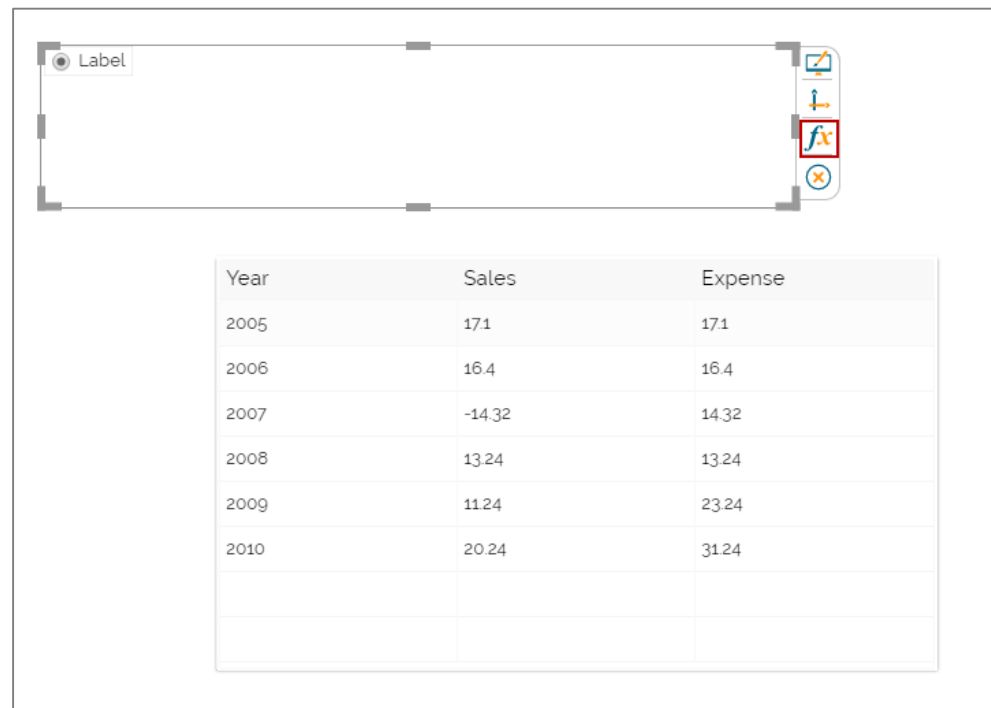
- Click 'Preview'.
- Click the 'Label' component via the preview screen.
- The selected styles will be applied as below:



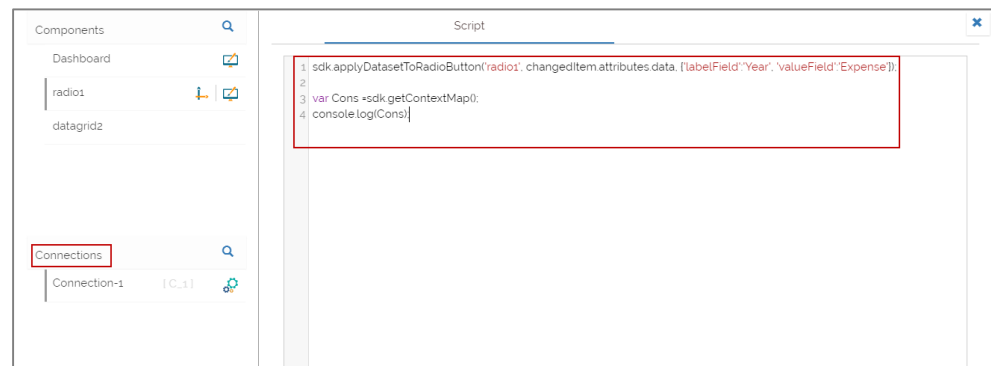
1.14.5. sdk.applyDatasetToRadioButton

This script will help to set data from the component to the Radio button.
The script can be explained through the following example:

- Drag and drop the required components onto the canvas. (In this case, the used components are: a 'Radio Button' and a 'Data Grid').
- Click the 'Script on Change' icon  of the any component to open the Scripting Window.



- Select 'Connections' option.
- Write the script as displayed in the below-given image:



- Click 'Preview'.
- Data of the 'Label' field will be displayed via the 'Radio' button.
- Users can view the following object details in the developer tools by clicking the F12 short-key.


```

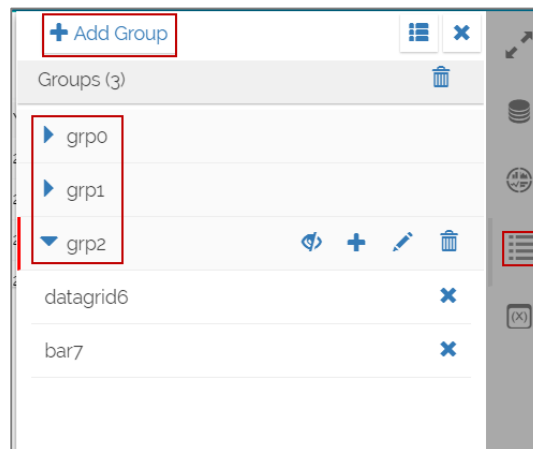
▼ {data: Array(5), FrameworkComponent: {...}} ⓘ
  ► FrameworkComponent: {dataSetValues: DataSetValues, dataObject: Array(5), status: {...}}
  ▼ data: Array(5)
    ► 0: {Year: 2008, Expense: 85, Revenue: 80}
    ► 1: {Year: 2012, Expense: 100, Revenue: 95}
    ► 2: {Year: 2013, Expense: 120, Revenue: 135}
    ► 3: {Year: 2014, Expense: 140, Revenue: 180}
    ► 4: {Year: 2015, Expense: 234, Revenue: 678}
    length: 5
    ► __proto__: Array(0)
  ► __proto__: Object
  >
  
```

1.14.6. sdk.hideGroup and sdk.showGroup

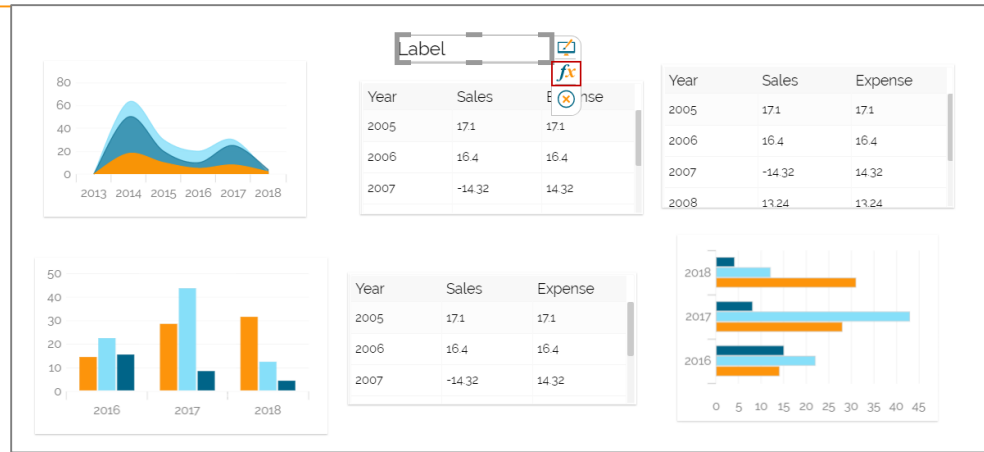
These scripts can be used to show and hide array of groups.

The script is explained via the following example of a dashboard:

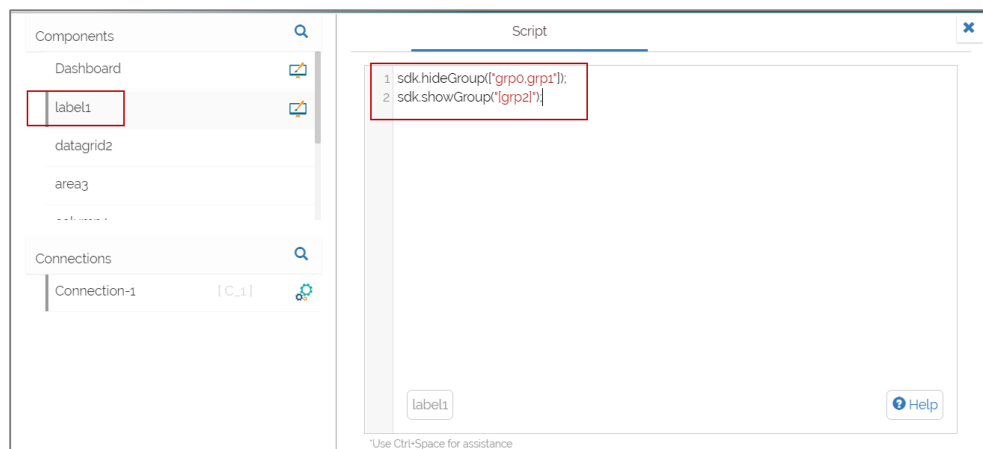
- Drag and drop the required components on the canvas to form groups (In this case the selected components are one 'Label', two 'Data Grids', and four 'Charts'-Area, Bar, Bubble, and Column).
- Click the 'Manage Dashboard Components' option 
- Users will be redirected to create groups and add the required components into a group.
 - In this case, the created groups are 'grp0', 'grp1' and 'grp2'.
 - Components added to the groups are as listed below:
 - grp0: area3, datagrid2
 - grp1: column4, datagrid5
 - grp2: datagrid6, bar7



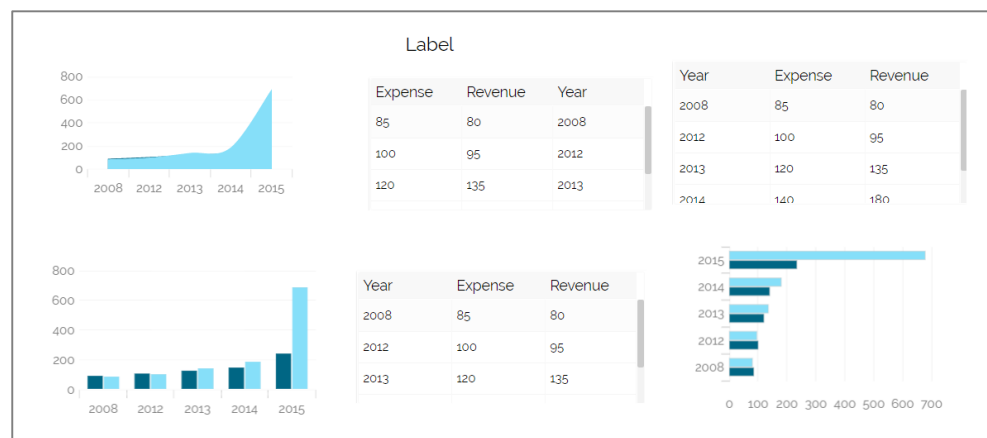
- Click the 'Script on Change' icon  of the 'Label' component to open the Scripting Window.



- Write the script as displayed in the following image:



- Click 'Preview'.
- Users will be redirected to preview the dashboard.



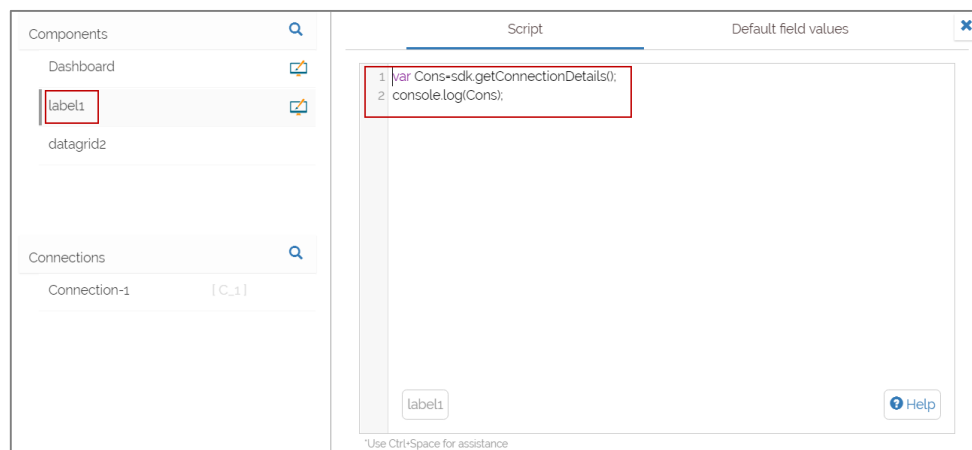
- Click 'Label'.
- The groups will be hidden or displayed based on the written script for the label.



1.14.7. sdk.getConnectionDetails and sdk.log

This script returns the connection details together with the log in the 'Console'.

- Drag and drop the required components on the canvas (In this case the selected components are a 'Label' and a 'Data Grid').
- Write the following script using the Label component:



- Click 'Preview'.
- Click the 'Label' component.
- Click 'F12' to open the developer's tools.
- The connection details will be displayed in the 'Console'.

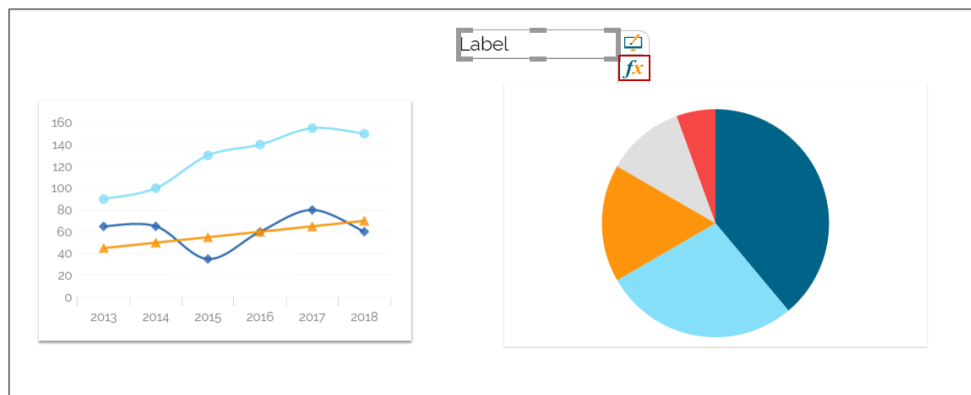

```

Console What's New
top Filter Default levels
▼ [{}]
▼ 0:
  Type: "offline"
  ▶ availableConditionSet: (3) [{}], [{}], [{}]
  ▼ availableFieldSet: Array(3)
    ▶ 0: {name: "Expense", displayName: "", type: "", added: true, $$hashKey: "object:17431"}
    ▶ 1: {name: "Revenue", displayName: "", type: "", added: true, $$hashKey: "object:17432"}
    ▶ 2: {name: "Year", displayName: "", type: "", added: true, $$hashKey: "object:17433"}
    length: 3
    ▶ __proto__: Array(0)
  dataSourceName: ""
  dataSourceType: ""
  databaseName: ""
  hostName: ""
  nividhQueryName: ""
  queryName: ""
  selectedDataSourceID: ""
  selectedServiceID: ""
  url: "DD Model Data.xlsx"
  ▶ __proto__: Object
  length: 1
  ▶ __proto__: Array(0)
  >
  
```

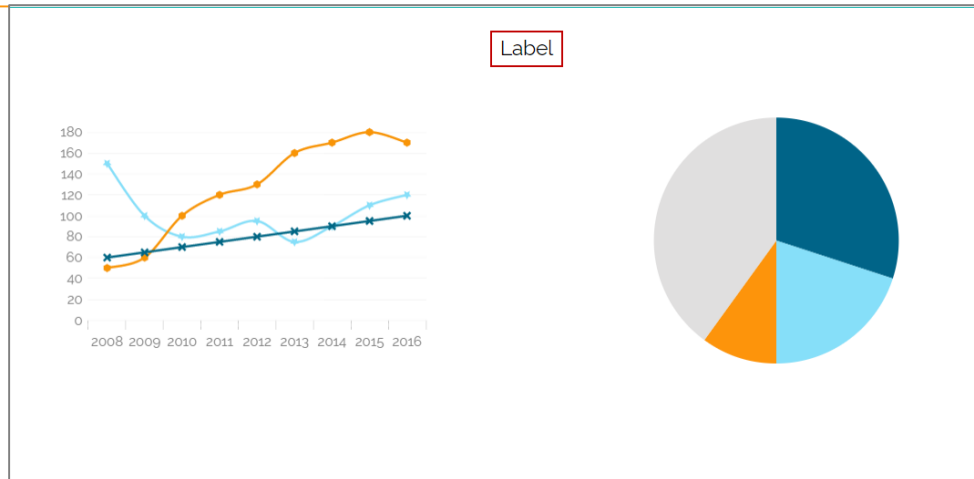
1.14.8. sdk.getDashboardJSON

This script will return the dashboard in JSON object.

- Drag and drop the required components on the canvas (In this case the selected components are a 'Label', two charts- 'Line' and 'Pie')
- Click the 'Script on Change' option using the Label component.



- Write the following component script using the Label component:
- Click 'Preview'.
- Click 'Label'.



- Click 'F12' to open the dashboard in JSON.

```

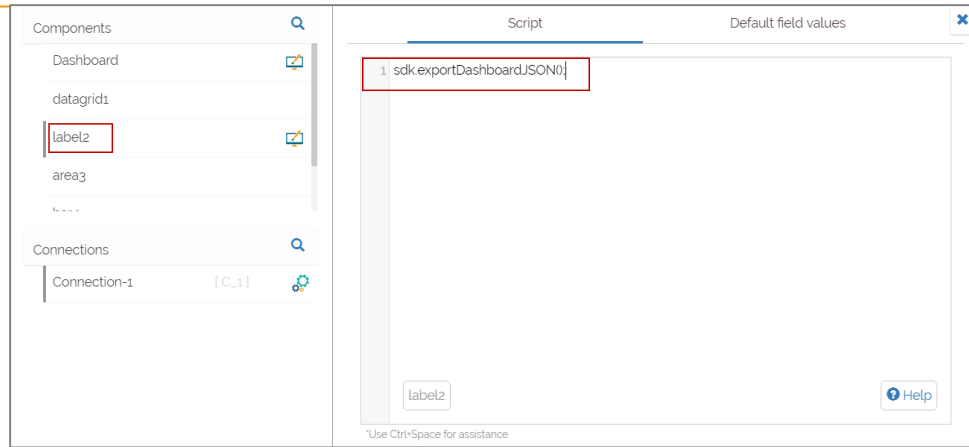
Console | What's New
top | Filter | Default levels
▼ {id: "dashboard201781364824794", backendId: "", workspaceId: "", DashboardOptions: {...}, versionDetails: {...}}
  ▶ AbsoluteLayout: {gradientRotation: "0", gradients: "#FFFFFF", bgAlpha: "1", borderThickness: 1, borderColor: "#08AEAB", ...}
  ▶ AuthProfiles: {Profile: Array(0)}
  ▶ DashboardOptions: {publishSQLServer: "", expiryDate: "", hasExpiry: ""}
  ▶ DataProviders: {DataURL: Array(2), queryServiceList: Array(0), selectedConnection: {...}}
  ▶ DatasetExpressions: {DatasetExpression: Array(0)}
  ▶ FlashVars: {Param: Array(1)}
  ▶ GlobalVariable: {Variable: Array(7)}
  ▶ OffLineData: {Data: Array(2)}
  ▶ VirtualDataSet: []
    backendId: ""
  ▶ componentGroups: []
    id: "dashboard201781364824794"
  ▶ imports: (3) ["Label", "LineChart", "PieChart"]
    importsStat: 1
    name: "Untitled Dashboard"
  ▶ variable: {Key: "dashboard", userScript: {...}, DefaultValues: {...}}
  ▶ versionDetails: {author: "BizViz Technologies Pvt. Ltd.", editedBy: "", lastUpdate: "", dashboardVersion: "3.0.0", publishDocId: "", ...}
  ▶ __proto__: Object
  >

```

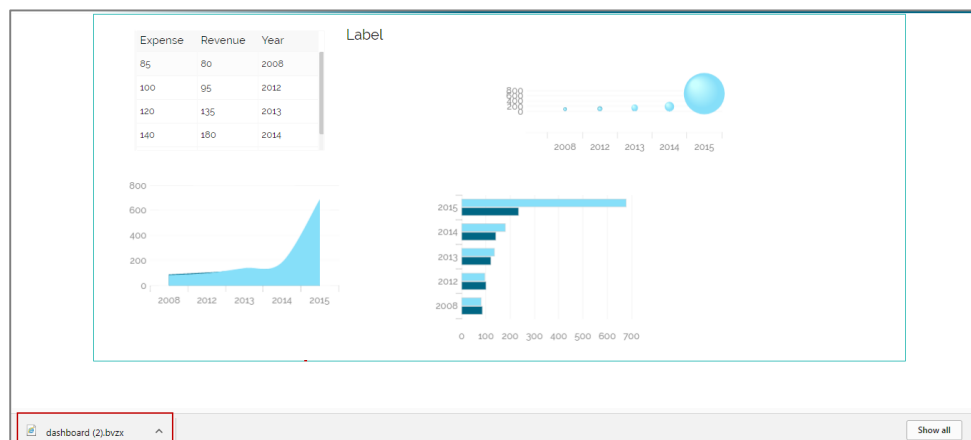
1.14.9. sdk.exportDashboardJSON

This script downloads the current dashboard in the BVZX file.

- Drag and drop the required components onto the canvas. (In this case, the used components are Label, Data Grid, and charts-Area, Bar, Bubble.
- Click the 'Script on Change' option via the Label component to open the scripting window.



- Write the script as mentioned in the following image:
- Click 'Preview'.
- Click 'Label' component on the preview screen.
- The selected dashboard will be downloaded to the system.

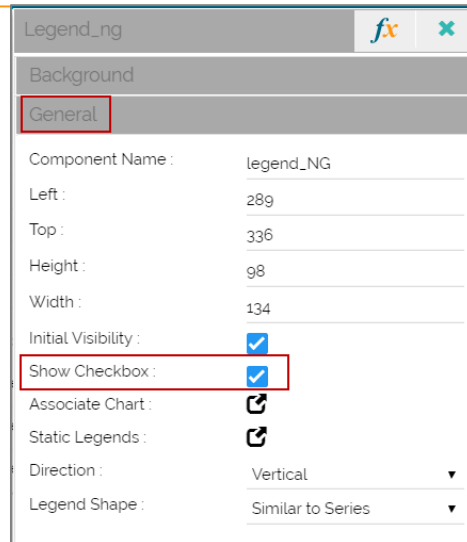


Note: Steps to open the exported BVZX file:

- Navigate to the dashboard designer home page.
- Click 'Open from the Local Disk'.
- Users need to select the path where the BVZX file has been downloaded to open it from the system.

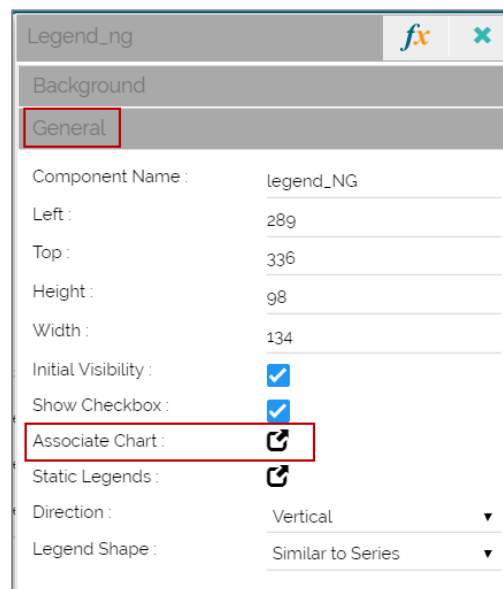
1.15. Associate Scorecard and Pivot Components with Legends

- Drag and drop the pivot grid and scorecard components onto the designer canvas and map them with a valid data source.
- Drag and drop the required Legend components onto the Designer canvas.
- Configure the following 'General' properties of a Legend Component.
 - Enable the checkbox by a checkmark in the box.

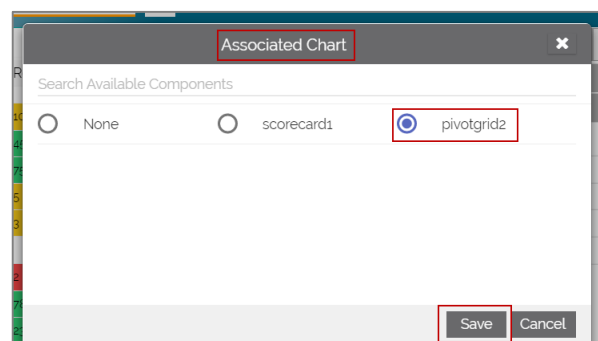


- Associate a chart:

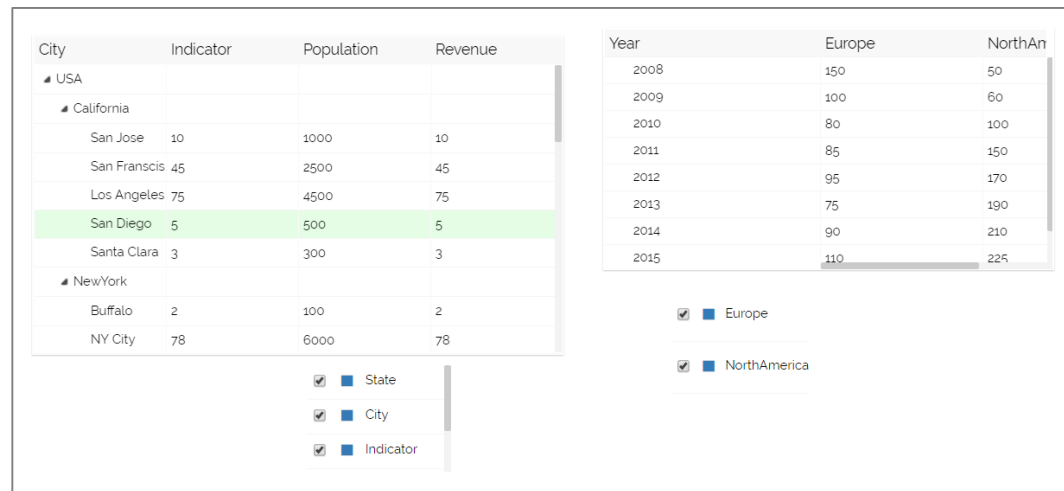
- Click the 'Associate Chart' option



- A new window will be opened displaying the dragged components on the canvas.
- Select a component by check marking the box.
- Click 'Save'.

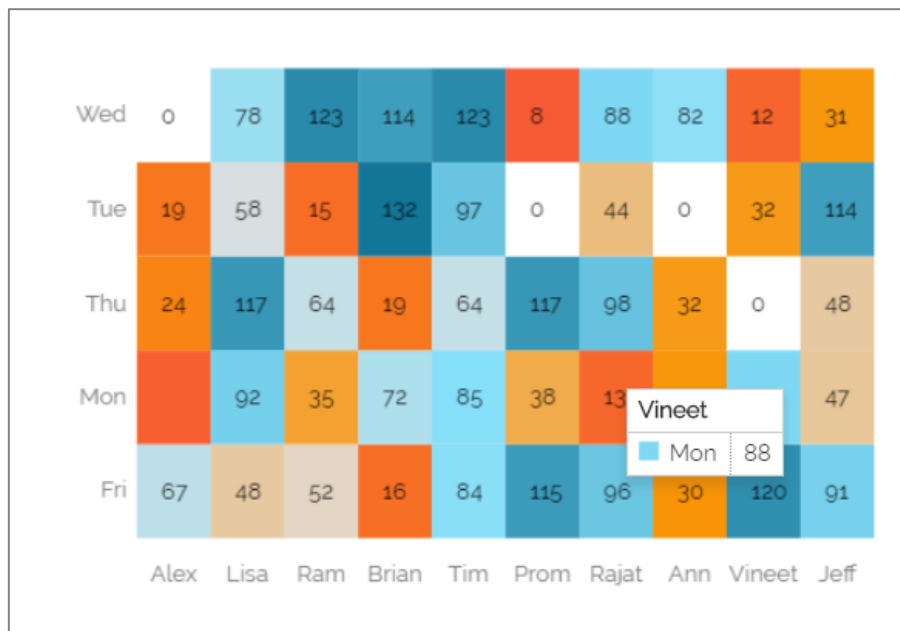


- Click 'Preview' to display the associated components with Legend.

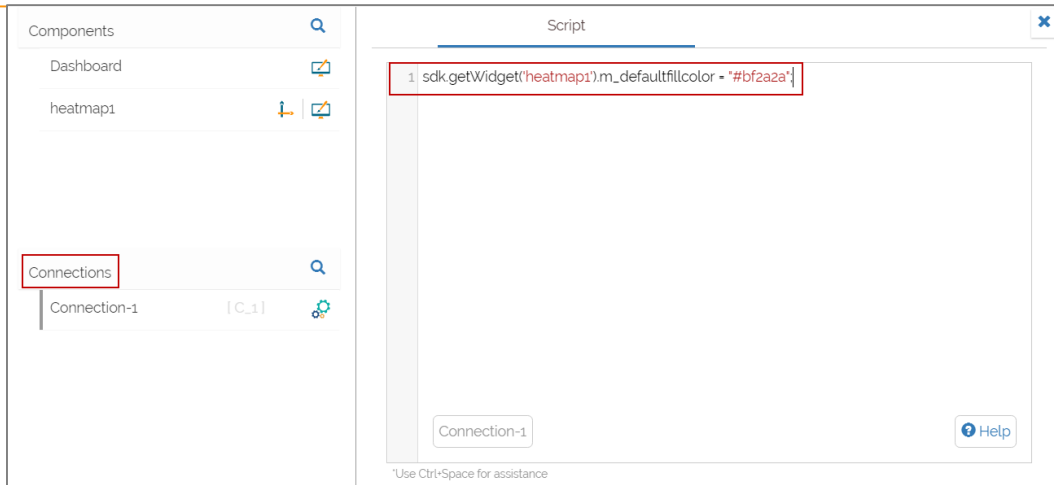


1.16. Heatmap chart: Color Enhancements

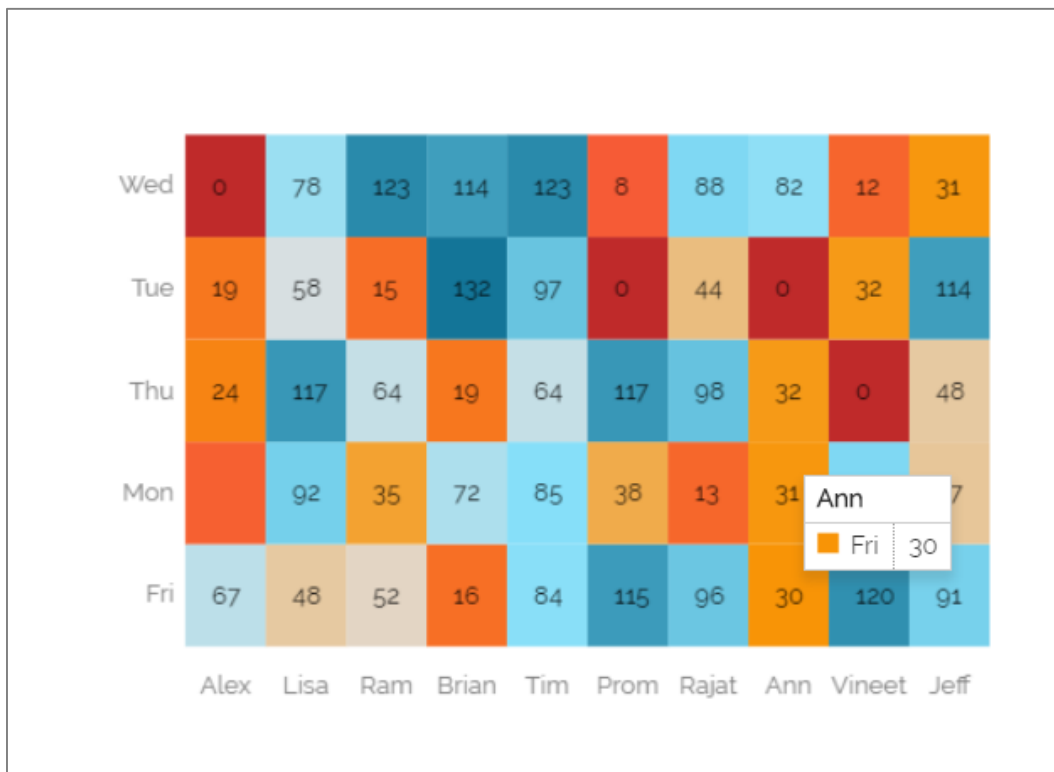
- Drag a Heatmap chart component onto the canvas.
- Map a data connection with the Heatmap chart component.
- Click 'Preview'.
- Null values will be displayed in white color by default in the preview.



- Users can change the color for those fields writing a connection script as shown in the following image:



- Click 'Preview' to see the selected color for the boxes containing the null values.



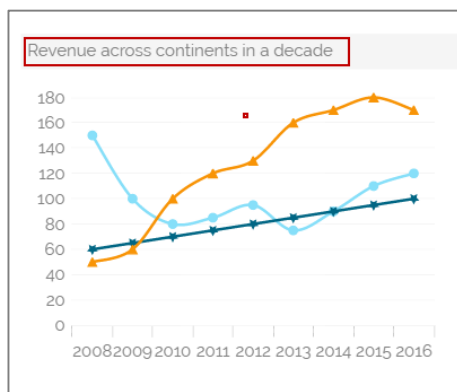
1.17. Chart Title Enhancement

Users can display any static string as a chart title along with global variable values or repeater field values.

- Open 'Title' properties fields.
- Describe about the chart using the 'Description' field.



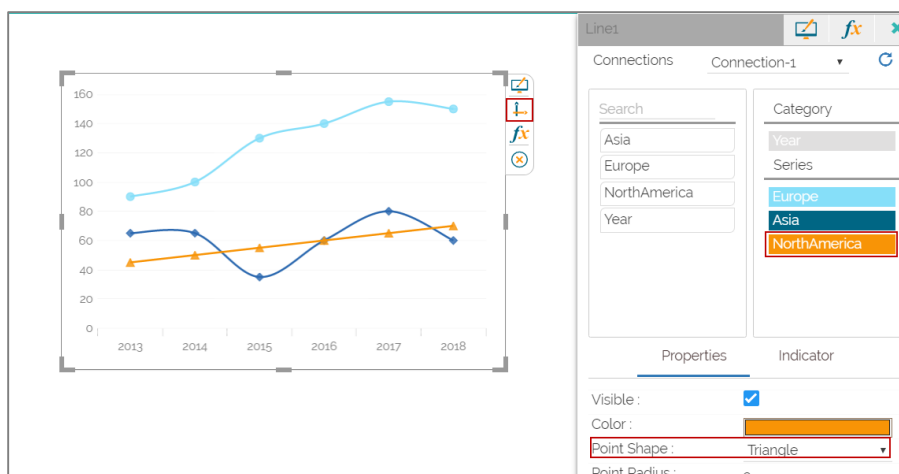
- The description will be added as title to the chart.



1.18. Enhancements in the Dataset Point Shape Properties

Legends and the tooltips of the related chart will display the selected 'Point Shape' for a field.

- Map a chart with a valid dataset.
- Configure the 'Point Shape' properties via the Dataset properties.



- Click 'Preview'.
- The tooltip will also display the same point-shaped.

