

Oracle Business Analytics

Oracle Data Visualization SDK

Antony Heljula
26 June 2017



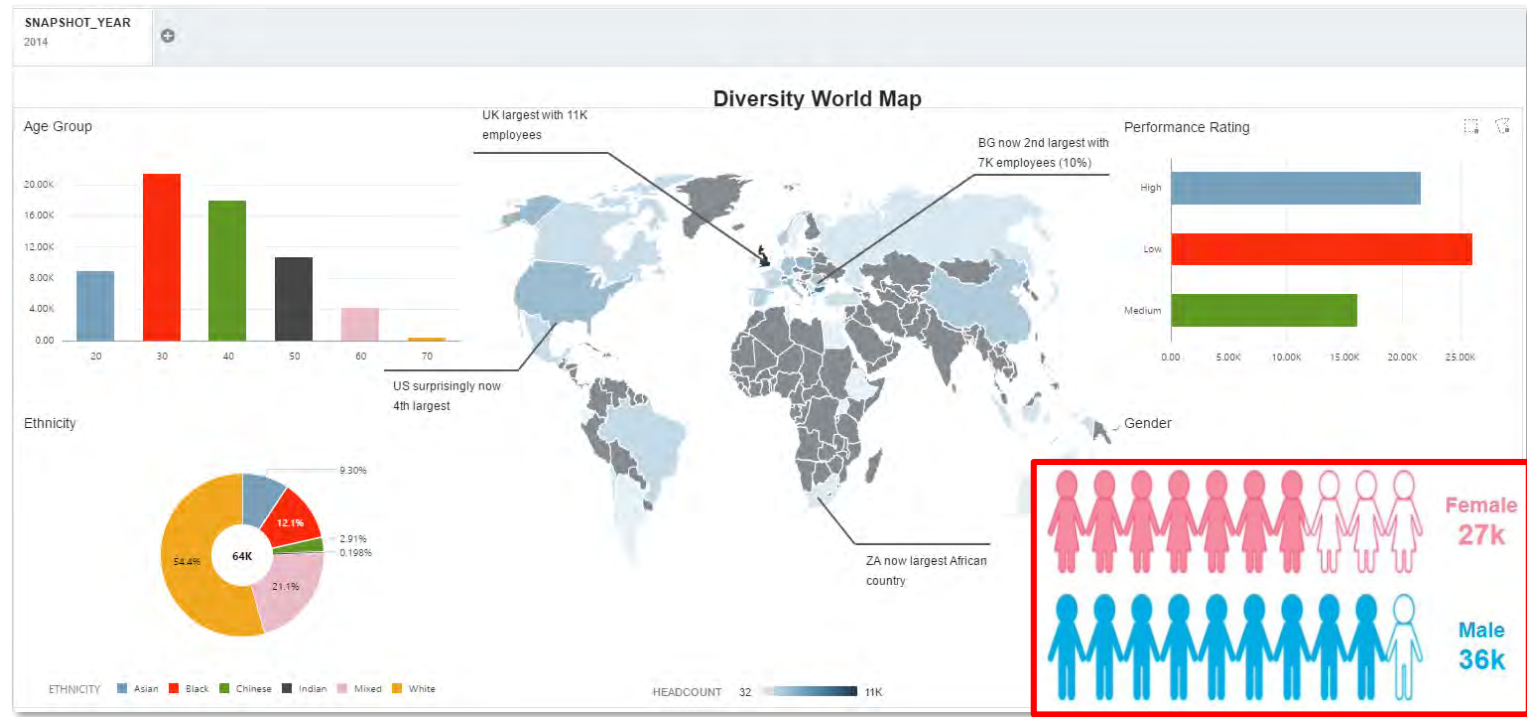
Oracle Data Visualization SDK Overview

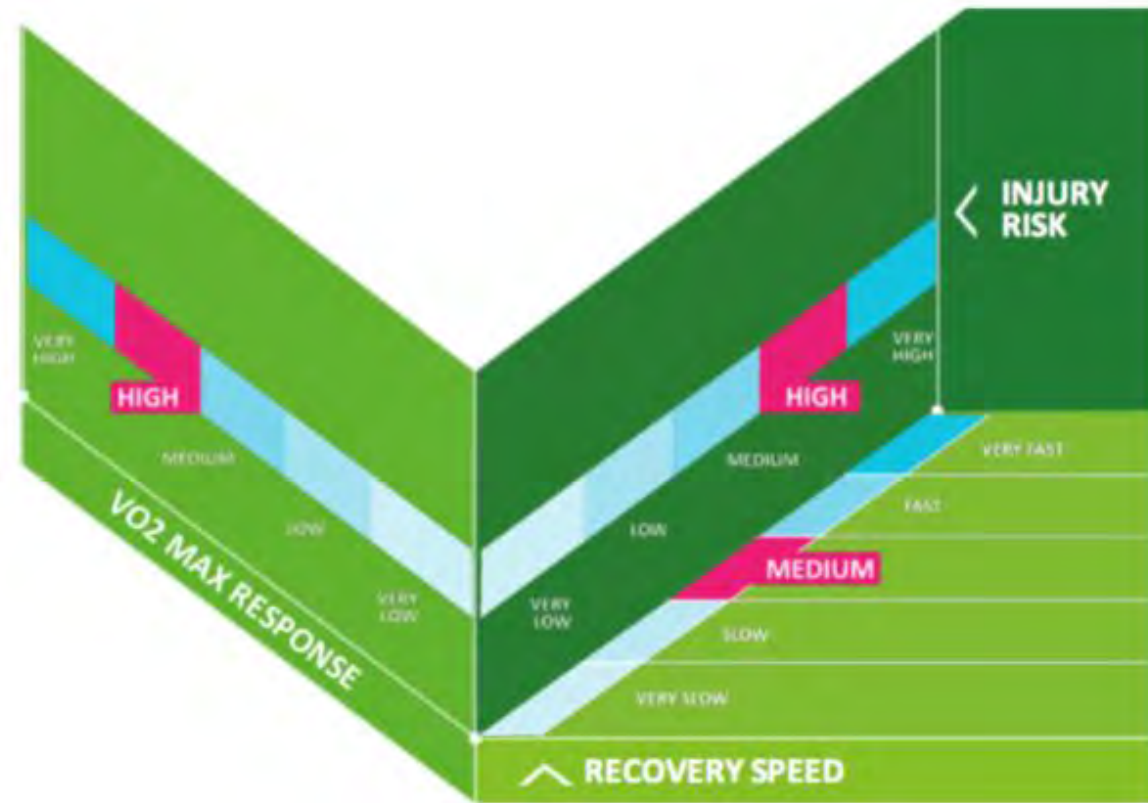
PEAK indicators

Overview

Oracle Data Visualization SDK

- The Oracle Data Viz SDK enables you to build your own “plugins” to implement custom visualizations
- Integrate a 3rd party chart engine or build any visualization you like!
- Requires knowledge of:
 - HTML
 - Javascript
 - CSS
 - JSON



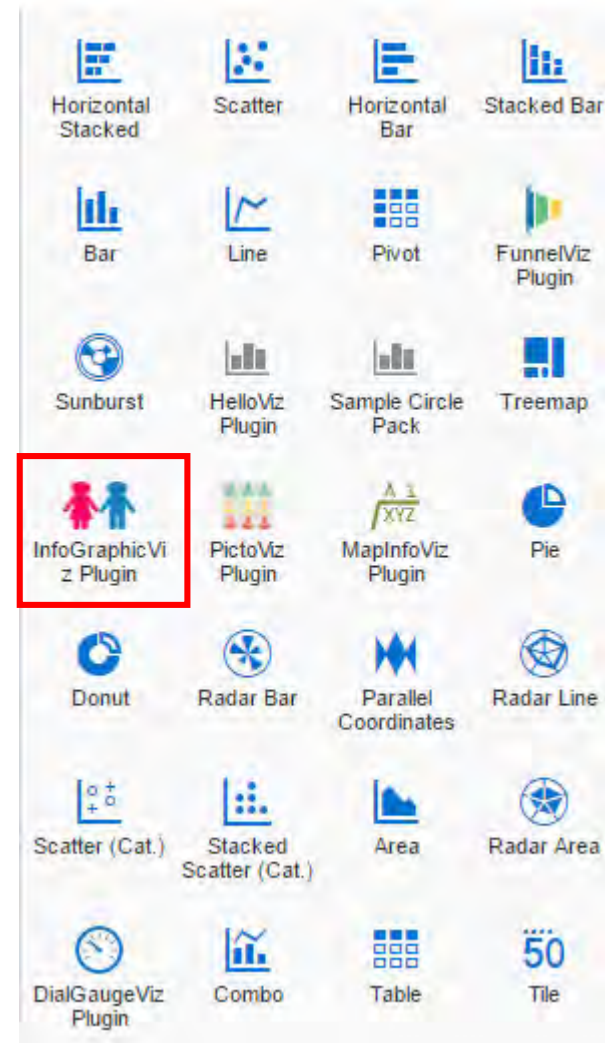


PEAK indicators

Overview

Oracle Data Visualization SDK

- Your custom visualizations can achieve the same level of interactivity and integration as the standard visualizations that ship with Oracle Data Viz
- They can be selected and used alongside all the standard visualizations
- Users should not need any additional training or technical know-how to embed your visuals
- This is a big difference/improvement compared to embedding 3rd party visuals within OBIEE



PEAK indicators

Overview

Oracle Public Store

- A public store is available where you can download plugins for Oracle Data Viz
- Your own plugins can also be made available here!

<http://www.oracle.com/webfolder/technetwork/OracleAnalyticStore/index.html>



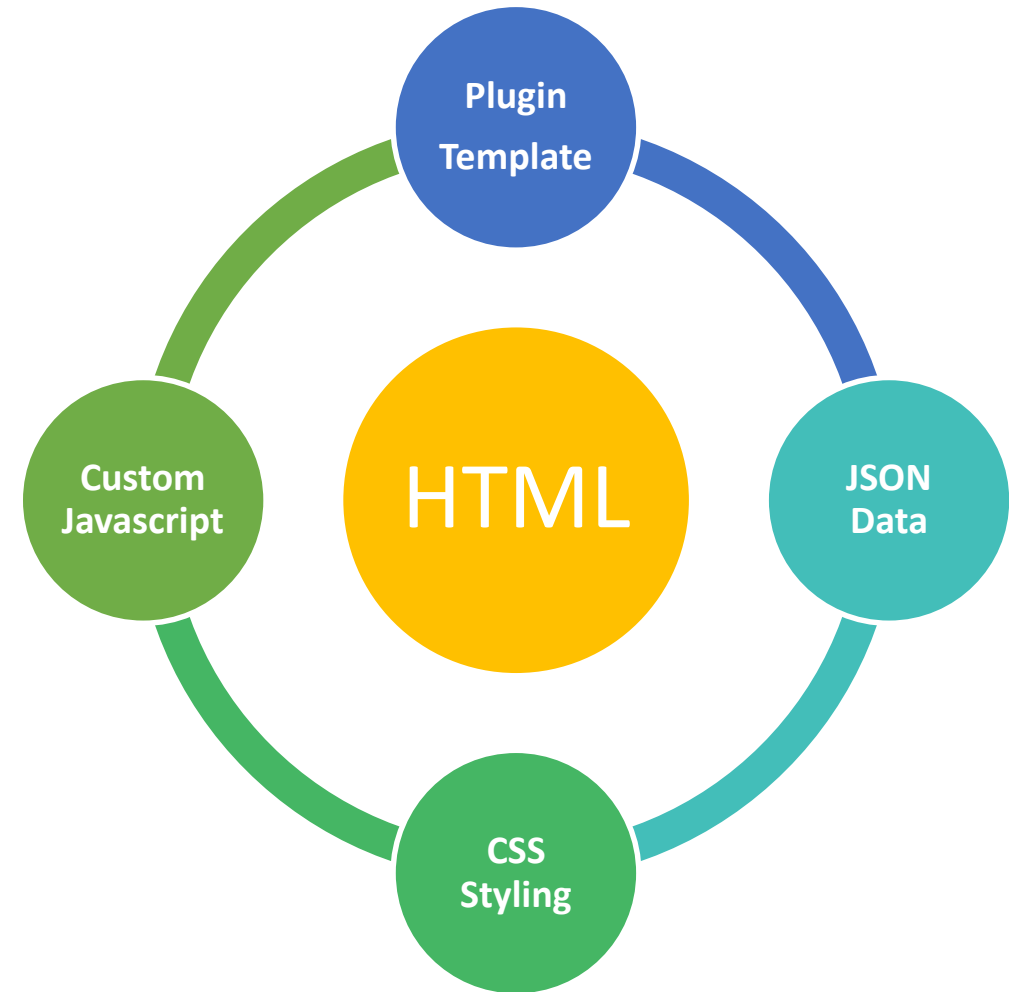
The screenshot displays the Oracle Analytics Store interface. At the top, it says "ORACLE Analytic Store" and "Oracle Analytics Store". Below this is a search bar and a "sort by Last Updated" dropdown. The main content area features five plugin cards:

- Funnel Viz Plugin**: OAC Data Viz, Sept 9, 2017. Description: "A custom viz plugin showing a Funnel Visualization to depict stages/phases within an overall process."
- Candlestick Viz Plugin**: OAC Data Viz, Sept 6, 2017. Description: "A custom viz plugin showing an interactive Candlestick Visualization most appropriate for stock-price like data."
- MapInfoViz-by Peak indicators**: Peak Indicators, May 12, 2017. Description: "This Map Info Viz plugin developed by PEAK indicators, enables you to overlay..."
- Collapsible Tree Plugin**: OAC Data Viz, May 5, 2017. Description: "The Collapsible Tree Plugin produces a 'node-link' diagram that lays out the connection between nodes in a way that..."
- Auto Refresh**: OAC Data Viz, Mar 30, 2017. Description: "The Auto Refresh custom viz plugin allows you to refresh your data in DV projects automatically in a given time interval. This..."

Overview

High-Level Components

- A javascript template is provided which you can extend with your own javascript code, CSS styling and images
- The template does enough to build a JSON array containing your data, plus it provides all the building blocks for interaction and integration with the product
- Your plugin essentially needs to generate HTML code that will be placed inside its own dedicated container (<DIV> tag) within the browser window



PEAK indicators

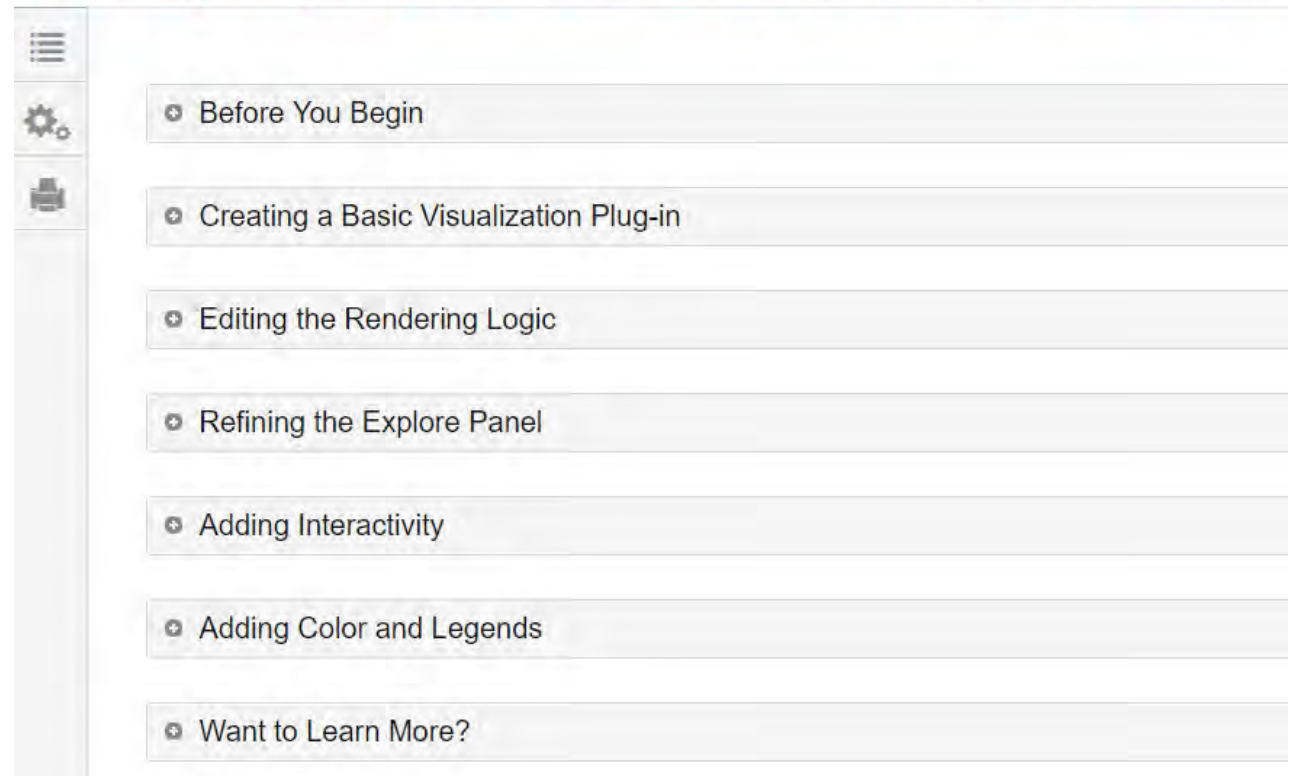
Overview

Oracle Tutorial

- Oracle provides a tutorial for building a plugin that uses the D3 chart engine
- The code examples are a useful starting point for integrating with D3
- Excellent support and encouragement also received from Oracle Development (including fixing code)

http://www.oracle.com/webfolder/technetwork/tutorials/obe/fmw/bi/dvdesktop/viz_plugin/dvd_sdk.html

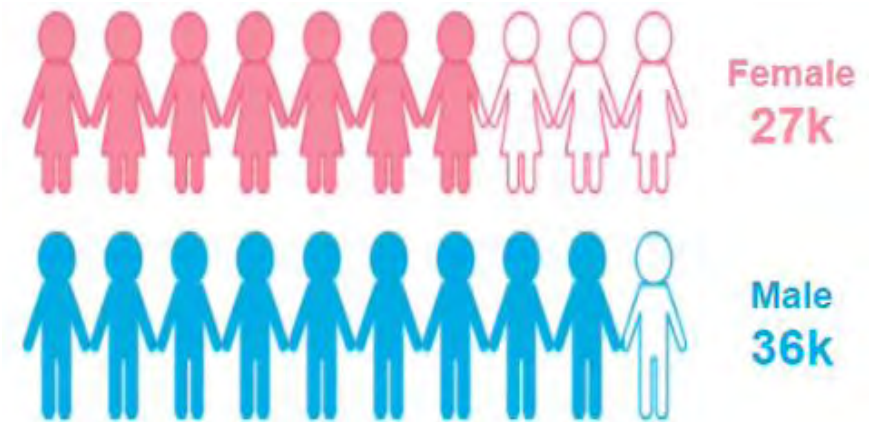
Creating Plug-in Visualizations for Oracle Data Visualization



Agenda

- Designing a Visualization
- Demonstration
- Getting Started with the SDK
- Building Your Visualization
- Brushing
- Marking
- Menu Options
- Custom Properties
- Data-Model Handler

InfoGraphicViz Plugin



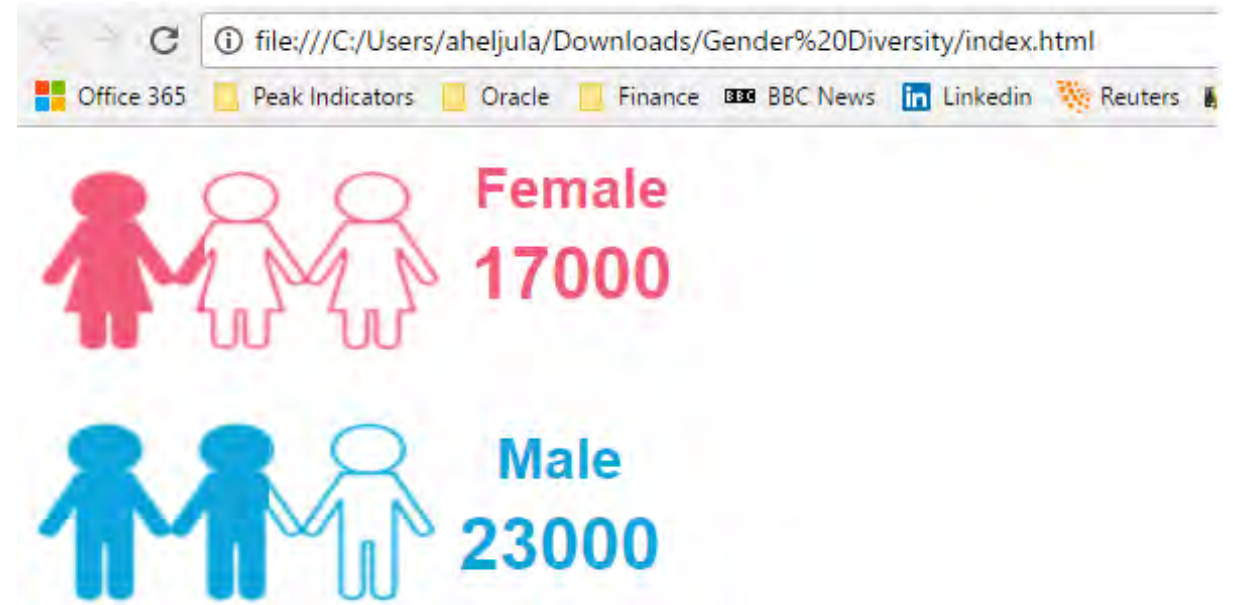
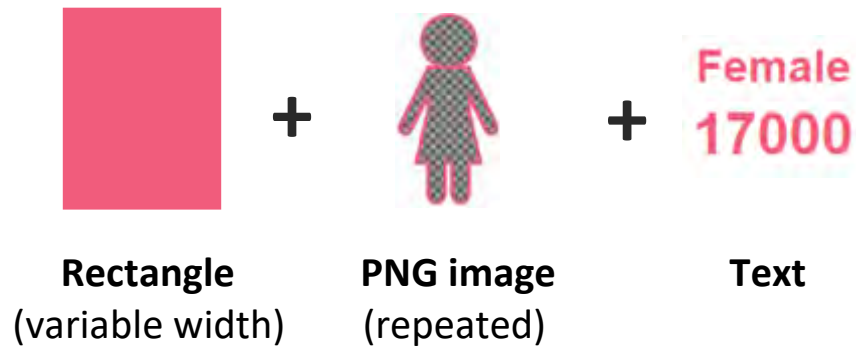
Oracle Data Visualization SDK

Designing a Plugin

PEAK indicators

Designing a Plugin Prototype / Mockup

- The first step is to produce a working mockup of the visualization that you wish to implement
- In our case, the mockup is built from 3 separate components:



Designing a Plugin Prototype / Mockup

- When building your mockup, try to make it reflect how it will work in Oracle Data Viz
- For example:
 - Use a JSON array to store the sample data
 - Use variables everywhere you can rather than hard-coding values, it helps you plan what properties the user will be able to adjust

```
//Define our data set
var data = {
  "name": "root",
  "children": [{
    "size": 23000,
    "id": ".Male",
    "name": "Male"
  },
  {
    "size": 17000,
    "id": ".Female",
    "name": "Female"
  }]
};

window.onload = function(){
  //For demonstration purposes fill the graphic on page load
  fillGraphic();
};

/*
 * Function to set the width of the coloured area behind the image
 */
function fillGraphic(){
  var female_size = 0;
  var male_size = 0;
  var total_size = 0;
  var img_width = 150;
}
```

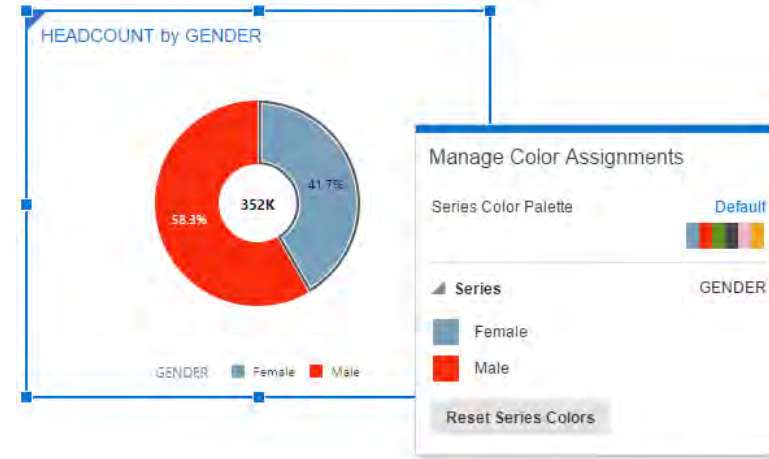
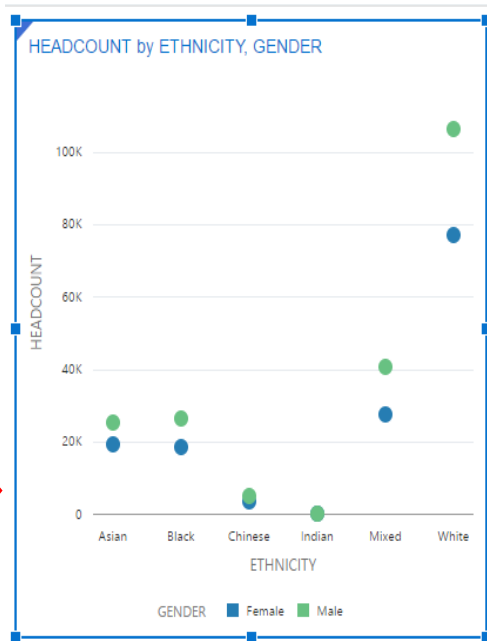
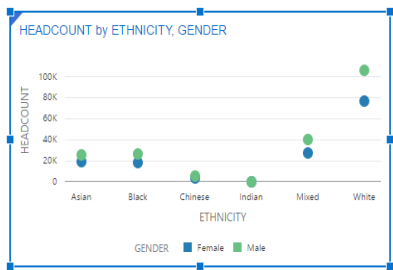
JSON Array

Variables for visualization properties

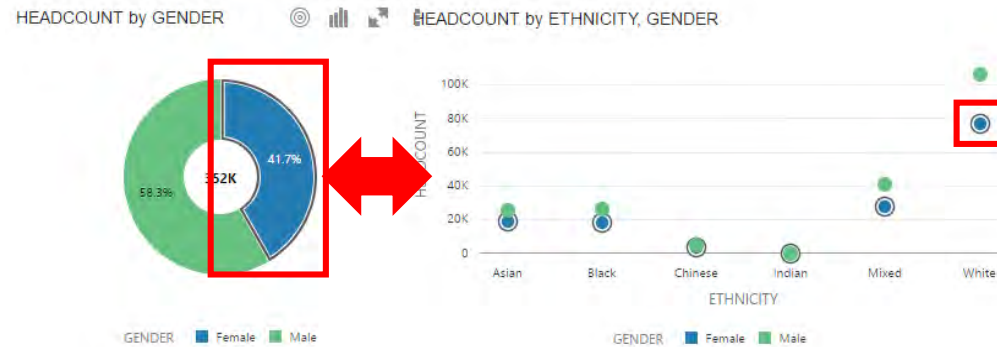
PEAK indicators

Designing a Plugin Standard DV Features

Freeform
Resizing



Colour
Assignment

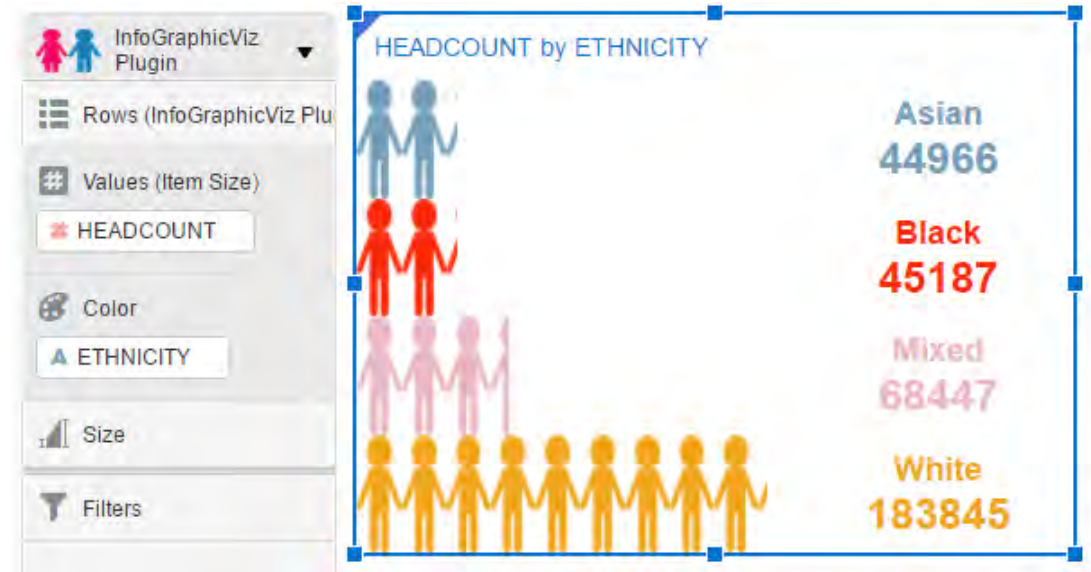


Brushing /
Marking

PEAK indicators

Designing a Plugin Other Features


- **Multiple rows** – you don't know how many rows of data you will have to present
- Your visualization has to cater for one or multiple rows, and the rows have to fit into the container:
- $\text{Row Height} = \text{Container Height} / \# \text{ Rows}$



PERFORMANCE_RATING	GENDER	HEADCOUNT
High	Female	52,211
	Male	68,975

Designing a Plugin Other Features

- **Nested JSON** – you have to plan for the JSON data structure being a “nested array” of data rather than your traditional “rows” of data
- In this example, you can see the rows on the table are actually sourced from a nested JSON structure
 - A parent JSON record for “High”
 - Two child JSON records for “Female” and “Male”
- With each child you get the following details:
 - The unique “**selectionId**” row and column – this is used to identify individual cells for the purposes of brushing and marking
 - The “**size**” attribute – this is the actual metric value
 - The “**color**” attribute – this is the color assigned in Data Viz
- The more dimensions/metrics you add, the more nesting that occurs!



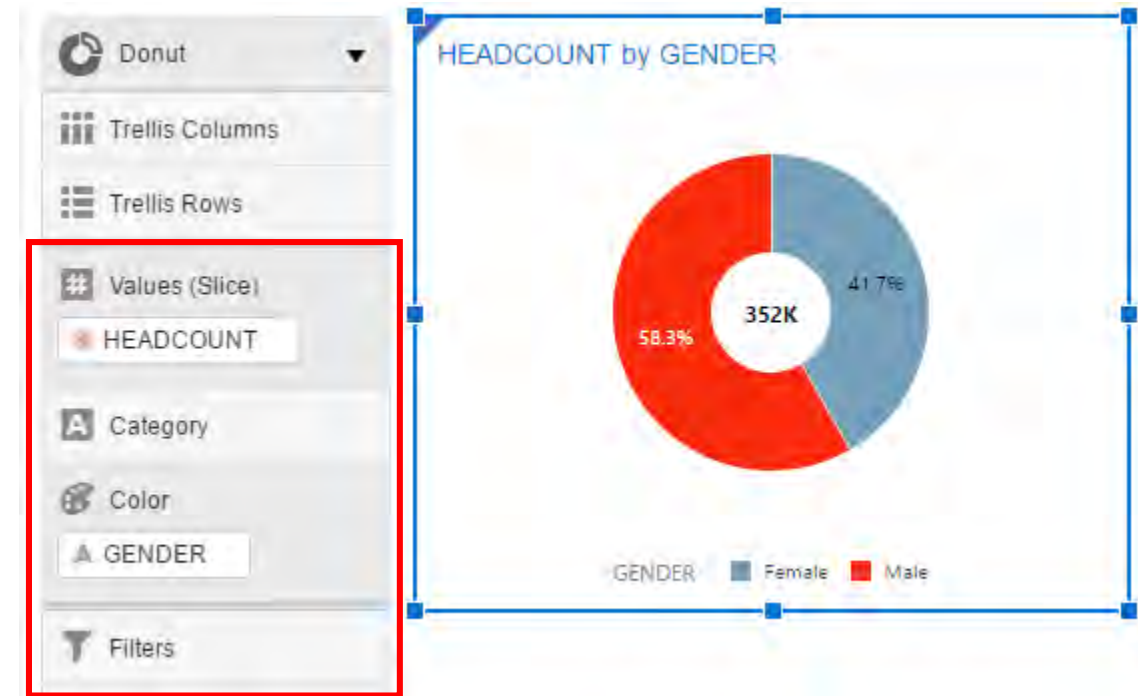
```
"name": "root",
"children": [{
  "size": 121186,
  "id": ".High",
  "name": "High",
  "children": [{
    "size": 52211,
    "id": ".High.Female",
    "name": "Female",
    "selectionID": {
      "row": 0,
      "col": 0
    },
    "color": "#98b8cc"
  },
  {
    "size": 68975,
    "id": ".High.Male",
    "name": "Male",
    "selectionID": {
      "row": 1,
      "col": 0
    },
    "color": "#4b7d9b"
  }
  ]
}
```

PEAK indicators

Designing a Plugin

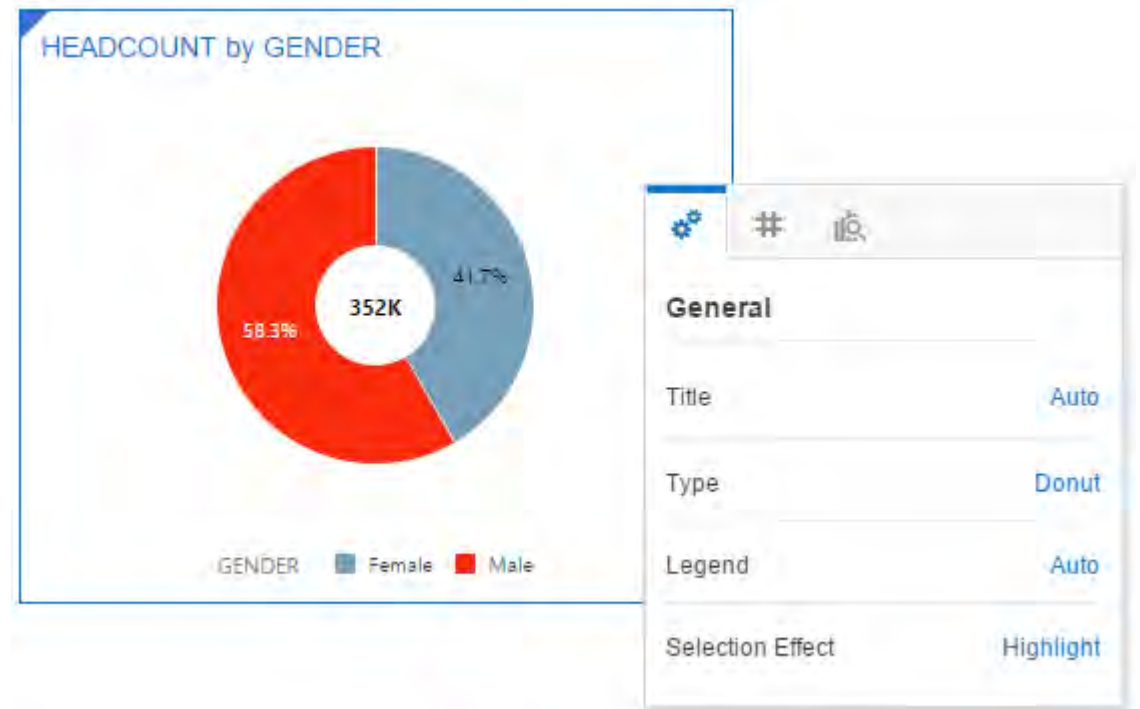
Other Features

- **Data Model Handler** – configuring the data objects used on the visualization
- You can change the labelling and also apply limits or disable features (this helps you restrict the amount of “nesting” that can occur in you JSON data!)
- For example:
 - No more than one object in “Values”
 - No more than one object in “Category”
 - Disable “Color”
 - Disable “Filters”
 - etc



Designing a Plugin Other Features

- **Properties** – each visualization has its own properties menu
- The plugin template automatically redraws your visualization
- What custom properties would you want your visualization to have?





Oracle Data Visualization SDK Demonstration

Oracle Data Visualization

InfoGraphicViz Plugin

Oracle Data Visualization SDK

Getting Started

Getting Started Creating Your First Plugin

- Install **Oracle DV Desktop**
- Create a plugin folder e.g.
`C:\Temp\OracleDV-Custom-Plugins`
- Then run the code on the right from an MS-DOS command window
- This will start up the SDK in your browser and create a new “**myDemoViz**” plugin
- NOTE: Comment out lines in red to start up SDK without creating a plugin

```
set DVDESKTOP_SDK_HOME=C:\Program Files\Oracle Data Visualization Desktop
set PLUGIN_DEV_DIR=C:\Temp\OracleDV-Custom-Plugins

set PATH=%DVDESKTOP_SDK_HOME%\tools\bin;%PATH%

mkdir %PLUGIN_DEV_DIR%
cd %PLUGIN_DEV_DIR%
bicreateenv
bicreateplugin viz -id com.company.myDemoViz -subType dataviz

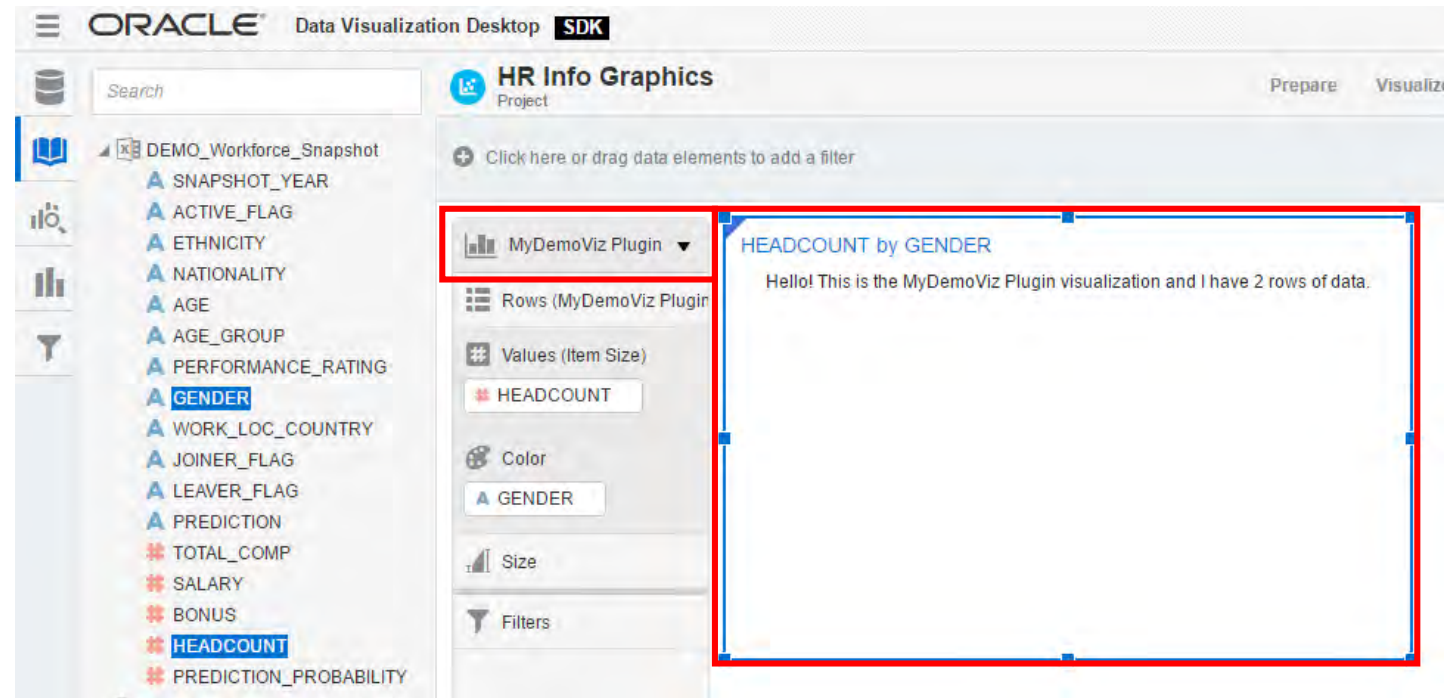
cd %PLUGIN_DEV_DIR%
.\gradlew run
```



PEAK indicators

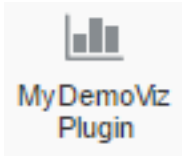
Getting Started Using Your Plugin

- Your new plugin is immediately available within the Oracle DV SDK!
- You get a default text message indicating the number of rows in the data returned



PEAK indicators

Getting Started Plugin Folder

- Your plugin folder will contain 3 key files
 - **myDemoViz.js** – the core javascript code
 - **myDemoVizIcon.png** – the default image icon 
 - **myDemoVizstyles.css** – your custom CSS styles

OS (C:) > Temp > OracleDV-Custom-Plugins > src > customviz

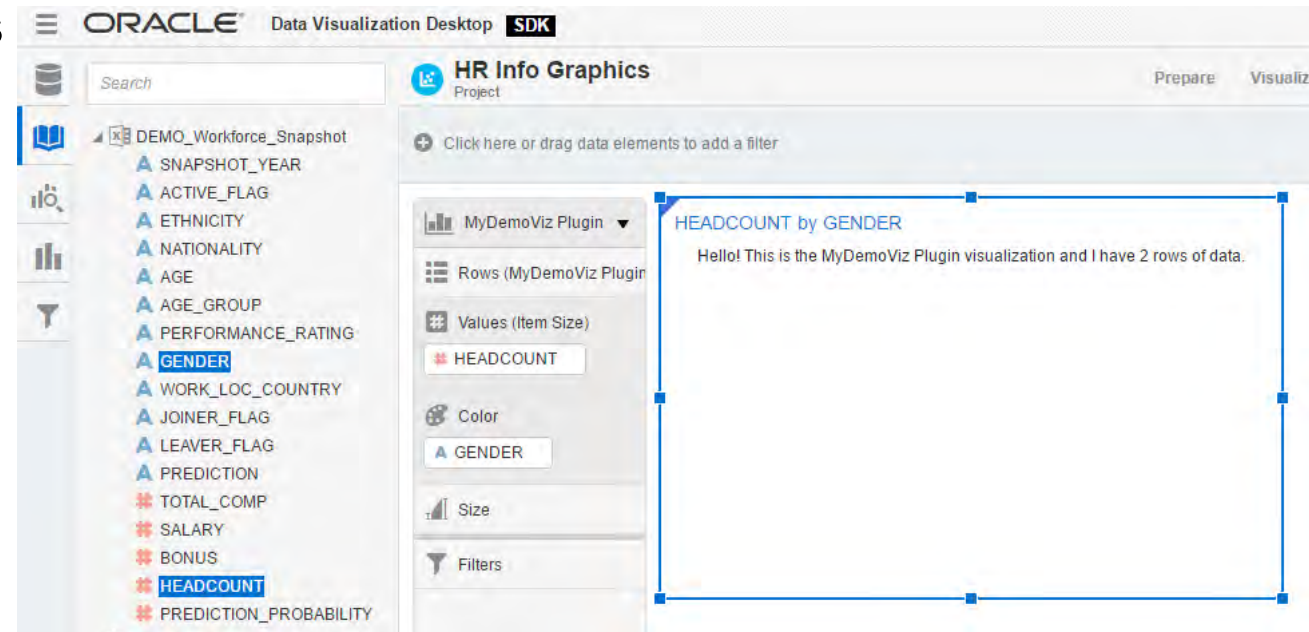
Name	Date modified
com-company-autoRefresh	19/04/2017 13:03
com-company-dialGaugeViz	19/04/2017 10:59
com-company-funnelViz	18/04/2017 13:01
com-company-helloViz	19/04/2017 22:44
com-company-infoGraphicViz	12/06/2017 21:15
com-company-mapInfoViz	08/06/2017 11:26
com-company-myDemoViz	24/06/2017 13:40
com-company-pictoViz	18/04/2017 12:24



extensions
nls
myDemoViz.js
myDemoVizdatamodelhandler.js
myDemoVizIcon.png
myDemoVizstyles.css

Getting Started Limitations

- Bizarrely, the default template plugin that is generated for you does not support a number of core features and you have to make quite a few modifications to enable them:
 - Colour selections
 - Brushing / marking
 - Custom properties
 - Colour attributes within JSON data
 - Standard interactions



Getting Started

Our Alternative Template Plugin

- Our alternative template is attached here, it can also be downloaded from our blog: <http://www.peakindicators.com/blog>
- If your plugin has a different name (e.g. **kScopeViz**) then you can still use this template, all you have to do is:
 1. Replace the contents of your plugin .js file with this one
 2. Search and replace “**MyDemoViz**” (case sensitive) with the name of your plugin e.g. **KScopeViz** (first letter uppercase)
 3. Search and replace “**myDemoViz**” (case sensitive) with the name of your plugin e.g. “**kScopeViz**” (first letter lowercase)

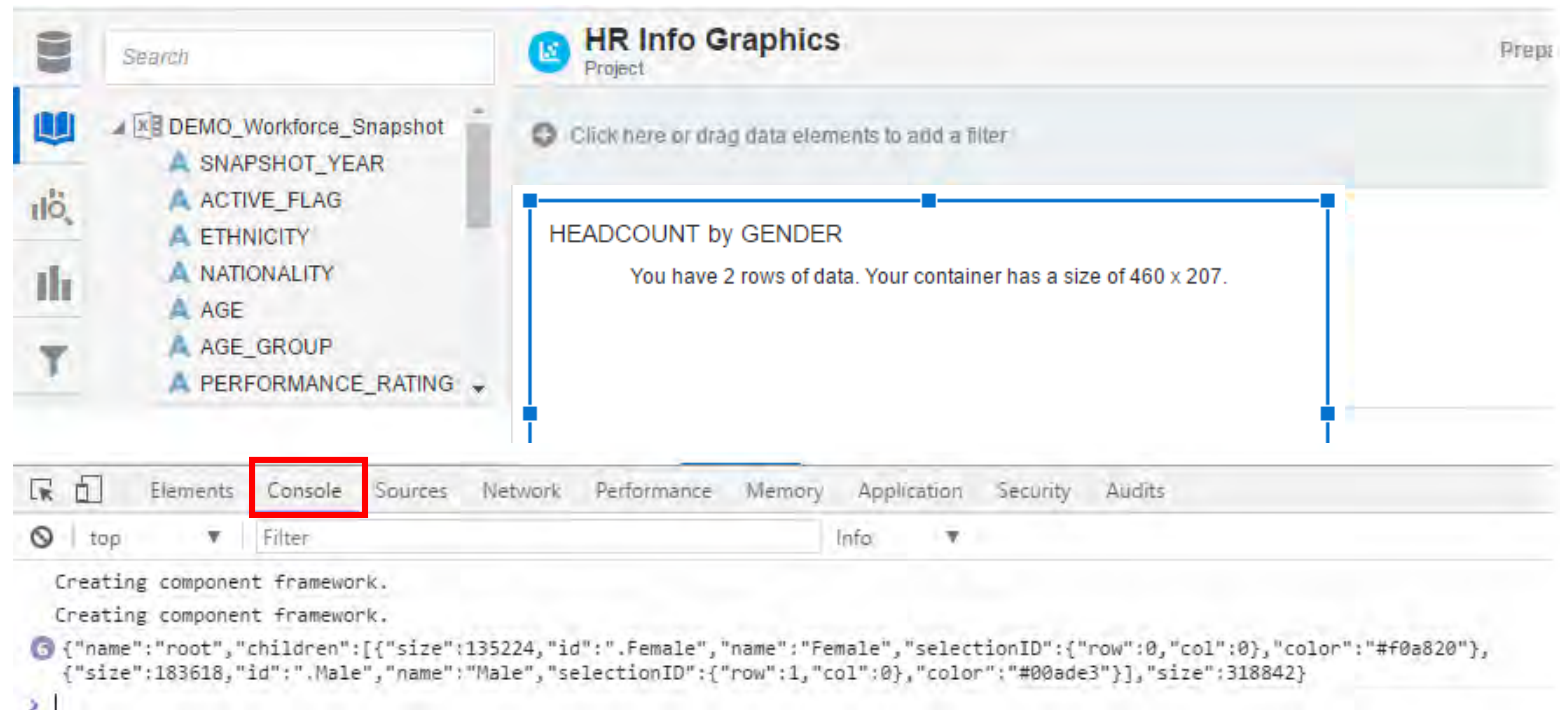


myDemoViz.txt

PEAK indicators

Getting Started Chrome Developer Tools

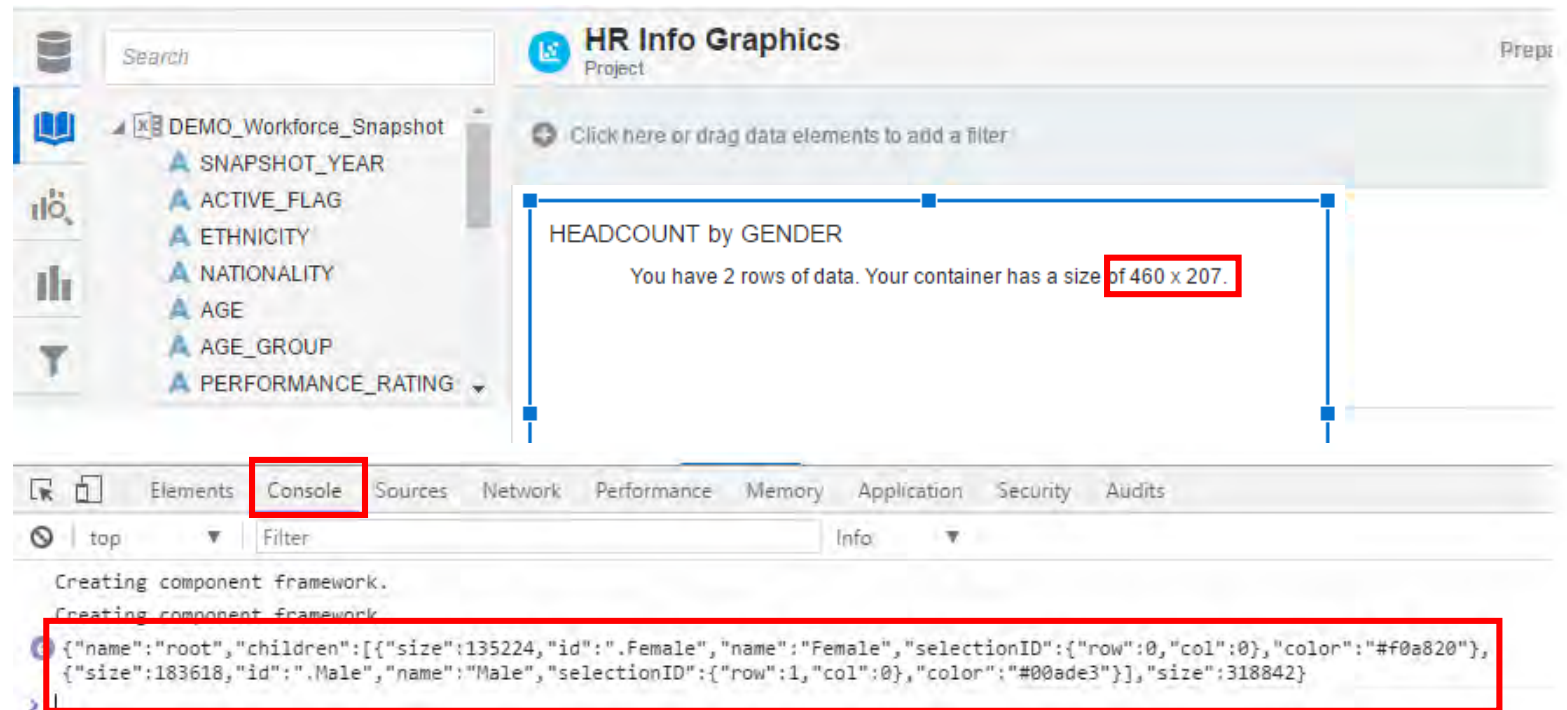
- The Chrome “Console” window will show any errors if your code fails to run – useful for debugging purposes
- You can access this console output simply by pressing F12
- If you modify your javascript, all you need to do is refresh your browser using <CTRL+F5>



PEAK indicators

Getting Started Chrome Developer Tools

- With our template plugin, you will by default see the JSON data logged in the Chrome “Console” window
- You now have everything you need to start building your visualization!
- If you resize the visualization you can see the height/width change instantly

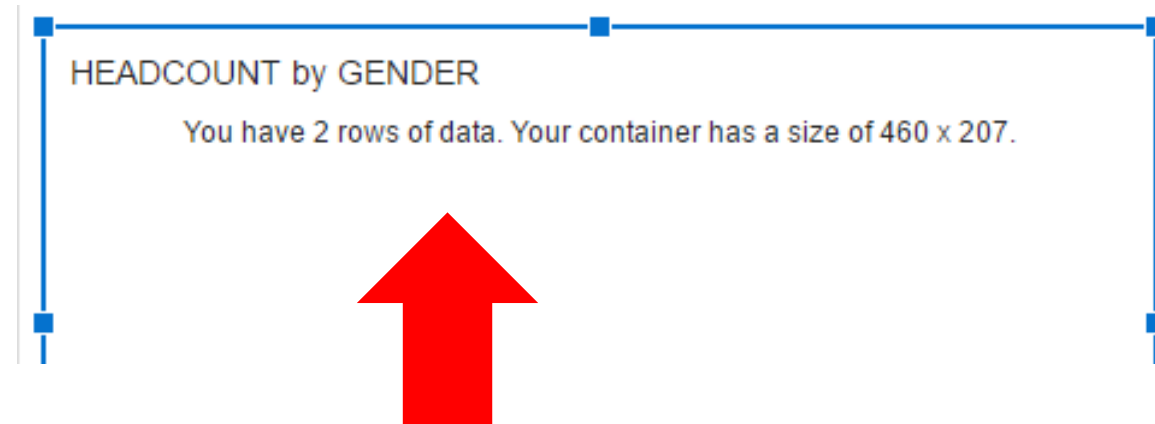




Oracle Data Visualization SDK Building your Plugin

Building your Plugin vizHTML

- Start with our alternative plugin (slide 31)
- As you can see, we build up a variable “**vizHTML**” to contain the final results to publish into our container
- Variables nRows, nWidth and nHeight defined for you



```
125 // Display the HTML within the container provided by Data Viz
126
127 var vizHTML = "You have " + nRows + " of data. Your container has a size of " + nWidth + " x " + nHeight + ".";
128
129 $(elContainer).html(vizHTML);
130
131 };
132
```

Building your Plugin Example: Building a table



1 - Building a table.txt

```
// Start to build our visualization. First of all initialize the vizHTML variables
// and also calculate the row height

var vizHTML = "";
var rowHeight = nHeight / nRows;

for (var i = 0; i < nRows; i++) {

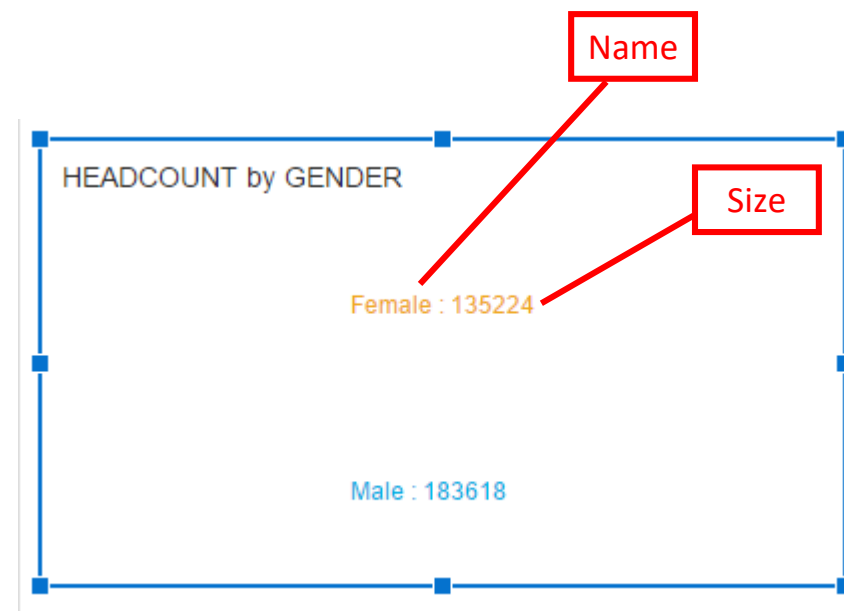
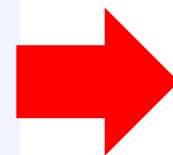
    var name      = oData.children[i].name;
    var size      = oData.children[i].size;
    var color     = oData.children[i].color;
    var selectionID = oData.children[i].selectionID;
    var row       = selectionID.row;
    var col       = selectionID.col;

    vizHTML = vizHTML + "<tr style=\"height:" + rowHeight + "px;\"> +
                        "<td style=\"color:" + color + ";\">" + name + " : " + size +
                        "</td><tr>";

}

var vizHTML = "<table>" + vizHTML + "<table>";

// Display the HTML within the container provided by Data Viz
$(elContainer).html(vizHTML);
```

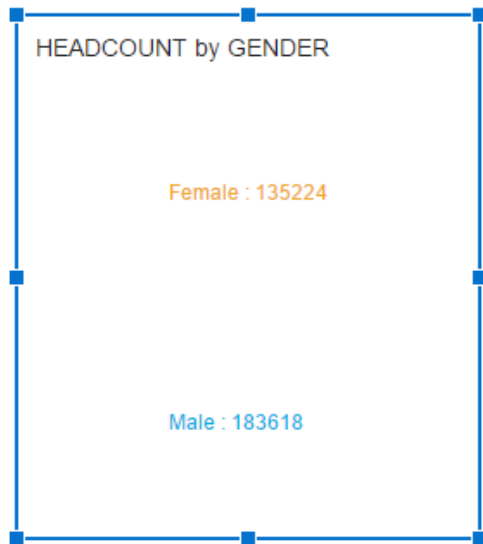


PEAK indicators

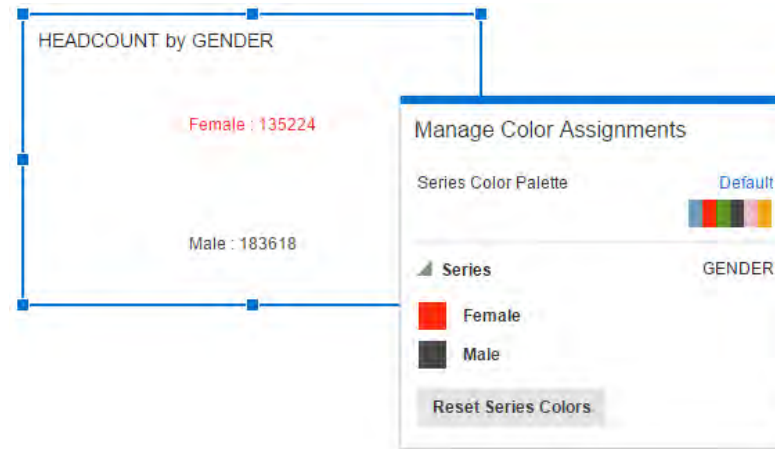
Building your Plugin Example: Building a table

- Note that the visualization already supports 3 important features

Freeform Resize



Colour Assignments



Switching Columns



PEAK indicators

Building your Plugin Example: CSS Styles

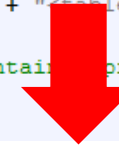


2 - CSS Styles.txt

- You can add CSS classes to your .css file and then reference them in your HTML
- In this simple example, we are using the “textDefault” class to set the background colour to grey and a bold font

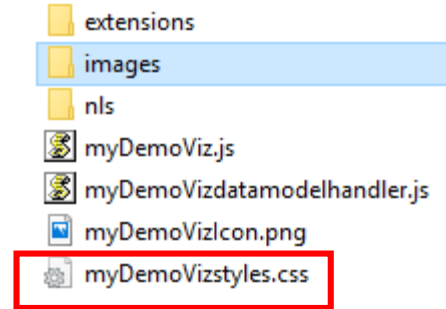
```
.textDefault {  
    background-color: #eeeeee;  
    font-weight:bold;  
}
```

```
// Start to build our visualization. First of all initialize the vizHTML var  
// and also calculate the row height  
  
var vizHTML = "";  
var rowHeight = nHeight / nRows;  
  
for (var i = 0; i < nRows; i++) {  
  
    var name      = oData.children[i].name;  
    var size      = oData.children[i].size;  
    var color     = oData.children[i].color;  
    var selectionID = oData.children[i].selectionID;  
    var row       = selectionID.row;  
    var col       = selectionID.col;  
    var rowClass  = "textDefault";  
  
    vizHTML = vizHTML + "<tr style='height:" + rowHeight + "px;'> " +  
        "<td class='" + rowClass + "' style='color:" + col  
        "</td><tr>";  
}  
  
var vizHTML = "<table>" + vizHTML + "</table>";  
  
// Display the HTML within the container provided by Data Viz  
$(elContainer).html(vizHTML);
```



Building your Plugin Referencing Images

- If you want to display images in your visualizations, then you can create an “images” folder in your plugin
- Note that the only way to use your images is to reference them via CSS classes. For example, to put a background image on a <DIV> tag:
 - <div class=“Male”>
- You cannot use tags in your generated HTML



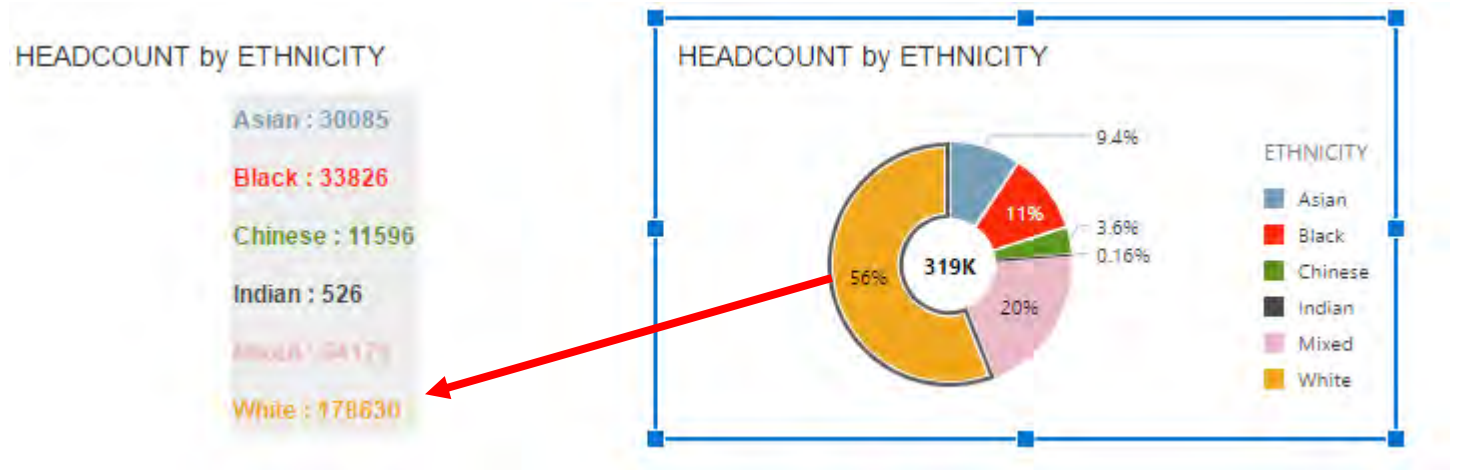
```
.Male {  
    background-image: url("images/Male.png");  
    background-size: 100% 100%;  
}  
  
.Female {  
    background-image: url("images/Female.png");  
    background-size: 100% 100%;  
}  
  
.Pound {  
    background-image: url("images/Pound.png");  
    background-size: 100% 100%;  
}  
  
.Dollar {  
    background-image: url("images/Dollar.png");  
    background-size: 100% 100%;  
}
```

Oracle Data Visualization SDK

Brushing

Brushing Overview

- We now want to configure our visualization to support **brushing**
- We want the visual to be highlighted when a related data item is selected in any other visualization



Brushing

Add CSS Style

- Add a new CSS style to use for any data items that required brushing
- In our example, we have a class “textSelected” that will have a dark background

```
.textSelected {  
    background-color: #333333;  
    font-weight:bold;  
}
```

PEAK indicators

Brushing Javascript



3 - Brushing.txt

- The code for brushing is very straight forward – the template already captures selected items from other visualizations
- The array “aSelectedItems” stores the list of data items to be brushed (a list of **row:col** combinations)
- We simply check if our data item exists in this array and use the alternate CSS class if it does

```
var rowClass      = "textDefault";

// Change class to textSelected if this data item has been selected
// in another visualization

var aSelectedItems = oViz.getSelectedItems();

if(aSelectedItems.indexOf(row+":"+col) >= 0)
    rowClass = "textSelected";

// Built the final vizHTML string

vizHTML = vizHTML + "<tr style=\"height:" + rowHeight + "px;\">" +
    "<td class=\"" + rowClass + "\" style=\"color:" + color + "\">" +
    "</td></tr>";

}
```

PEAK indicators

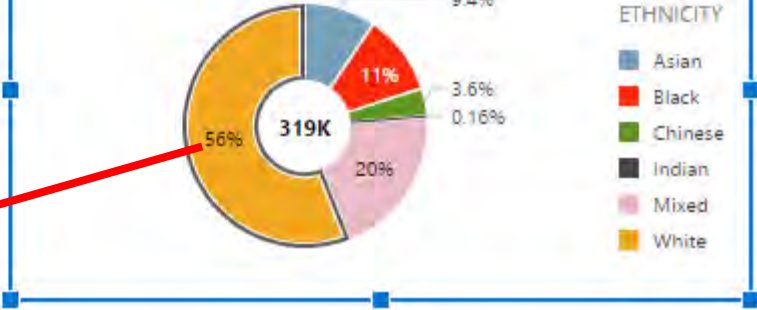
Brushing The Result

- Brushing now works!

HEADCOUNT by ETHNICITY

Asian : 30085
Black : 33826
Chinese : 11596
Indian : 526
Mixed : 68874
White : 178630

HEADCOUNT by ETHNICITY



Oracle Data Visualization SDK

Marking

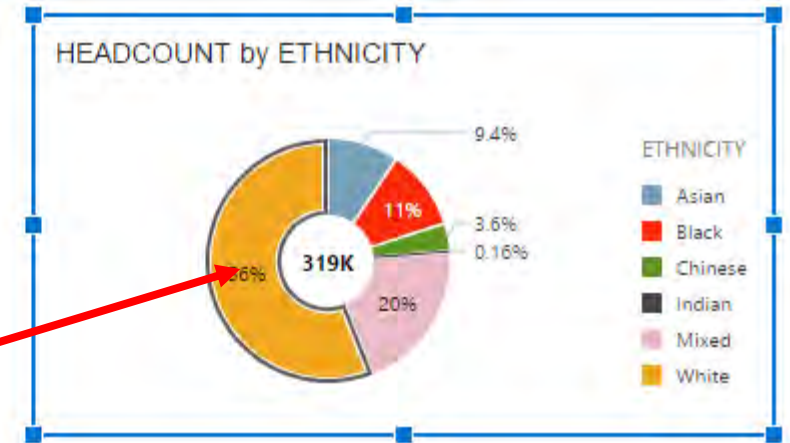
PEAK indicators

Marking Overview

- We want to initiate a “**marking**” event whenever a user selects a data item on our visualization
- The marking event will automatically trigger brushing on other visualizations

HEADCOUNT by ETHNICITY

Asian : 30085
Black : 33826
Chinese : 11596
Indian : 526
Mixed : 16174
White : 178630



PEAK indicators

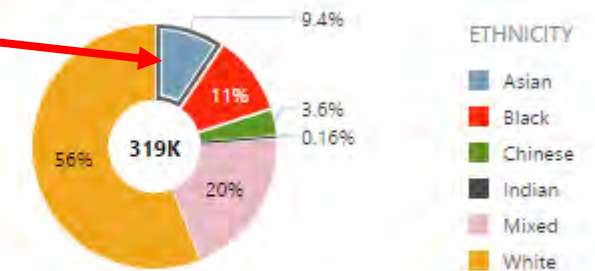
Marking The Result

- Our marking now works as expected!
- Note that when we “mark” one of our data items it automatically gets “brushed” at the same time, so it gets highlighted in the same way as when brushing occurs

HEADCOUNT by ETHNICITY

Asian : 30085
Black : 33826
Chinese : 11596
Indian : 526
Mixed : 64174
White : 178630

HEADCOUNT by ETHNICITY



Oracle Data Visualization SDK

Custom Properties

PEAK indicators

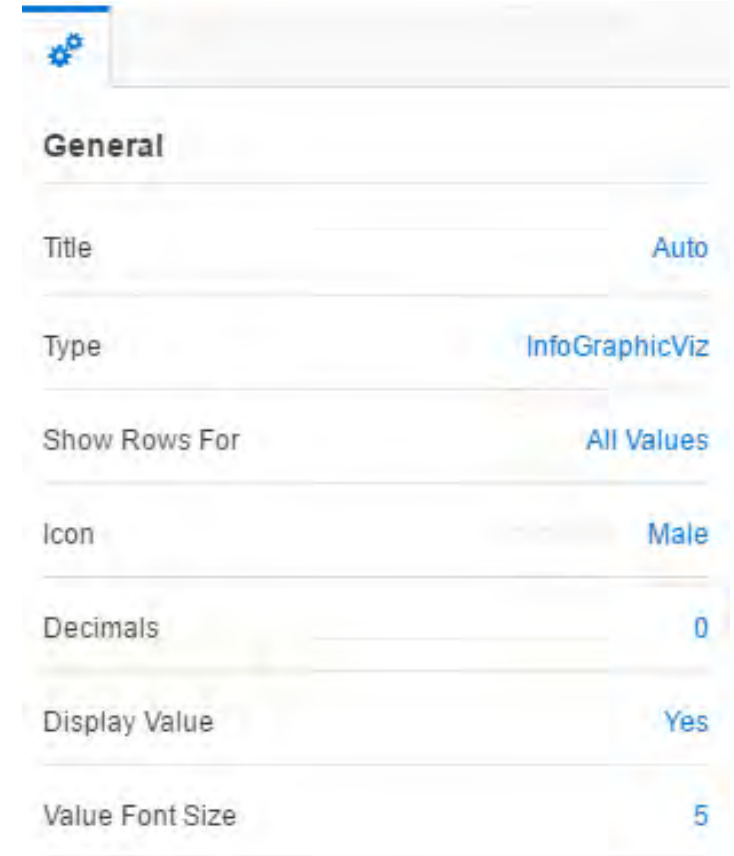
Custom Properties Overview

- By default, our visualization will only have two basic properties:
 - Title
 - Type (switch to another type of visualization)
- We can add our own custom properties, there are two types of property we can add:
 - A free text field e.g. **width of the table**
 - A drop-down menu e.g. **left/center/right text align**



Custom Properties Overview

- Adding custom properties requires a little bit more effort – we have to add code/functions in several places to handle:
 - Defining properties
 - Loading properties
 - Saving properties
 - Getting properties
 - Updating properties
- **Our plugin templates simplifies this by providing 10 pre-defined custom properties that you can configure**



General	
Title	Auto
Type	InfoGraphicViz
Show Rows For	All Values
Icon	Male
Decimals	0
Display Value	Yes
Value Font Size	5

Custom Properties

Step 1

- In your main function near the top, there is a placeholder to set up your custom properties
- As you can see, there are 10 properties configured, you just have to allocate the ones you need and set default values.
For example:
 - **property1** – table width (default 300px)
 - **property2** – text align (default center)

```
// Define custom properties for the visualization with defaults
// Further functions for loading and setting the properties

this.Config = {
  property1 : '300',
  property2 : 'center',
  property3 : '',
  property4 : '',
  property5 : '',
  property6 : '',
  property7 : '',
  property8 : '',
  property9 : '',
  property10 : ''
};
```

Define here you list of properties and their default values

Custom Properties

Step 2

- Add code to create your custom property menu items to function: **doAddVizSpecificPropsDialog**
- The code to add varies depending on whether you want a free-text field or a drop-down list
- This is the code to set “**property1**” as a free-text field with display text “**Width**”

```
// Configure your custom properties here
//

// Free text field for property1
var oGadgetFactory = this.getGadgetFactory();
var oGadget = new gadgets.GadgetValueProperties(eudef.GadgetTypeIDs.TEXT_FIELD, this.Config.property1);
var oGadgetInfo = oGadgetFactory.createGadgetInfo("property1Gadget", 'Width', 'Width', oGadget);
generalPanel.addChild(oGadgetInfo);
```

Custom Properties

Step 2

- And here is the code to set “**property2**” with display text “**Text Align**” and with 3 possible choices (left, right, center)

```
// Drop-down list for property2
var aValues = [];
aValues.push(new gadgets.OptionInfo('left', 'left'));
aValues.push(new gadgets.OptionInfo('right', 'right'));
aValues.push(new gadgets.OptionInfo('center', 'center'));

var oGadgetFactory = this.getGadgetFactory();
var oGadget = new gadgets.GadgetValueProperties(eudef.GadgetTypeIDs.TEXT_SWITCHER, this.Config.property2);
var oGadgetInfo = new gadgets.TextSwitcherGadgetInfo("property2Gadget", 'Text Align', 'Text Align', oGadget, eudef.GD_FIELD_ORDER_GENERAL_LINE_TYPE, false, aValues);
generalPanel.addChild(oGadgetInfo);
```

Array of choices

Custom Properties

Step 3



5 - Custom Properties.txt

- You can now reference your custom properties within your code that generates the HTML

```
// Obtain custom properties
var width = this.Config.property1;
var align = this.Config.property2;
```

Define variables "width" and "align" from custom properties 1 and 2

```
// Built the final vizHTML string

vizHTML = vizHTML + "<tr style=\"height:" + rowHeight + "px;text-align:" + align + ";\">" +
    "<td dvId=\"" + dvId + " dvRow=\"" + row + " dvCol=\"" + col + " class=\"" + rowClass + "\" +
    \" style=\"color:" + color + ";width:" + width + "px;\">" + name + " : " + size +
    "</td></tr>";
```

PEAK indicators

Custom Properties Test

- That's it! You can now test out your custom properties...



General

Title Auto

Type MyDemoViz Plugin

Text Align right

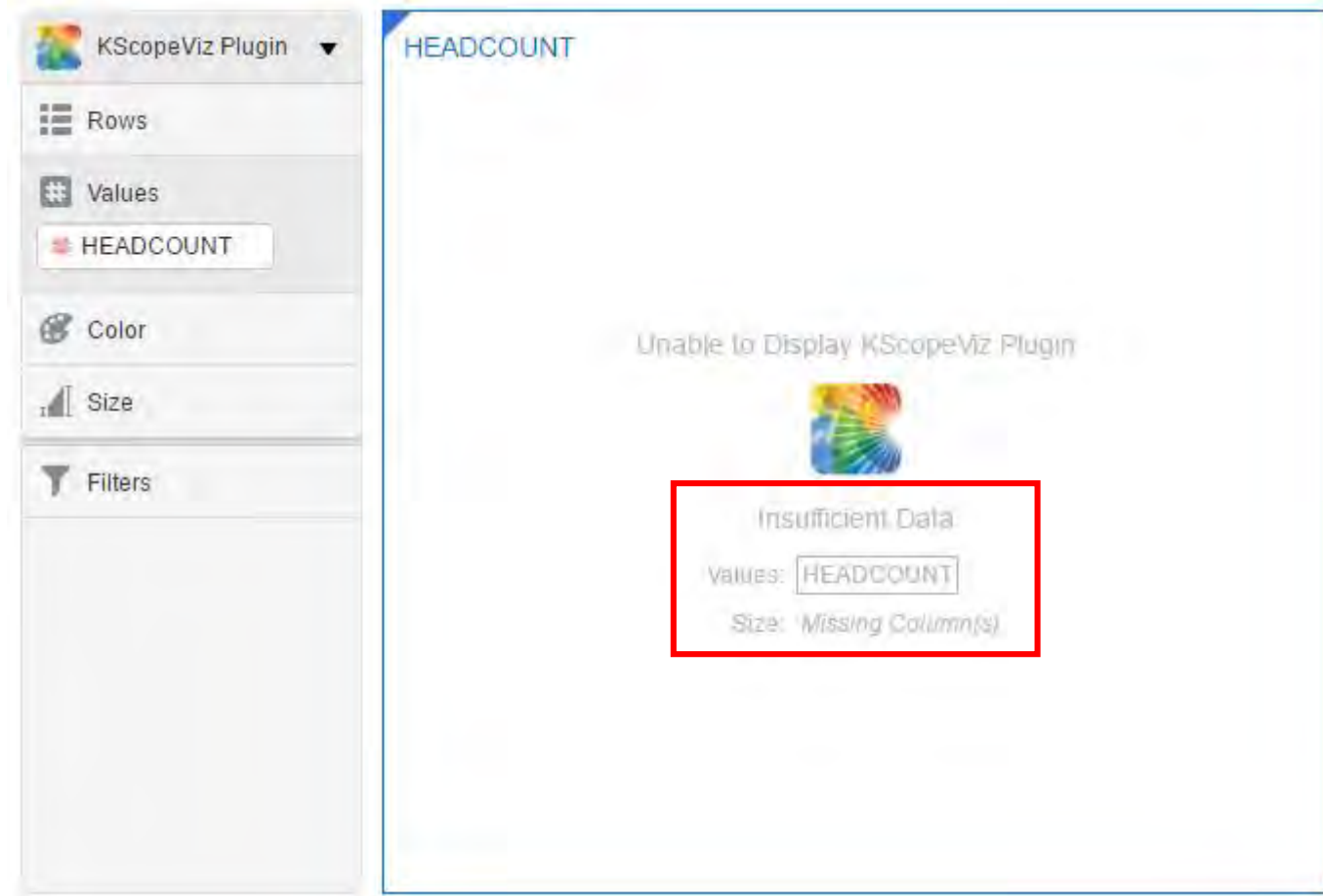
Width 200

Oracle Data Visualization SDK

Data-Model Handler

Data-Model Handler Overview

- The **data-model handler** is a JSON file that contains the data-model requirements covering Rows/Values/Color/Size sections including:
 - Min / Max number of objects
 - Priority
 - Disable
- In the example shown, Data Viz is returning an error to say that we have not provided enough data items to meet the data-model requirements of our visualization



The image shows a software interface for the KScopeViz Plugin. On the left is a configuration sidebar with sections for Rows, Values, Color, Size, and Filters. The 'Values' section is active, showing a dropdown menu with 'HEADCOUNT' selected. On the right is a visualization area titled 'HEADCOUNT'. It displays an error message: 'Unable to Display KScopeViz Plugin' with a colorful circular icon. Below this, a red-bordered box highlights the error details: 'Insufficient Data', 'Values: HEADCOUNT', and 'Size: Missing Column(s)'.

Data-Model Handler Editing

- The data-model handler file can be found in the following location in your plugin folder:
 - /extensions/oracle.bi.tech.plugin.visualization
DatamodelHandler
- In our example, we have made config changes to the “color”, “size” and “row” sections
- NOTE: After you have edited the file you must shutdown the SDK and restart it again in order for the changes to take effect

```
..
"row" : {
  "contentType": "mixed",
  "global": {
    "priority": 10,
    "preferredMax": 1
  },
  "ui": {
    "customName": { "key": "MAPINFOVIZ_ROW_LABEL", "default": "MapInfoViz Plugin Row" }
  },
  "categorical": {
    "functionalInfo": ["inner", "row", "categoricalType"]
  }
},
"color" : {
  "contentType": "both",
  "global": {
    "minCount": 1,
    "maxCount": 1,
    "preferredMin": 1,
    "preferredMax": 1,
    "priority": 7
  },
  "measures": {
    "maxCount": 1
  },
  "categorical": {
    "functionalInfo": ["inner", "col", "categoricalType"]
  }
},
"size": "none",
"graph": "none",
"detail": "none"
```

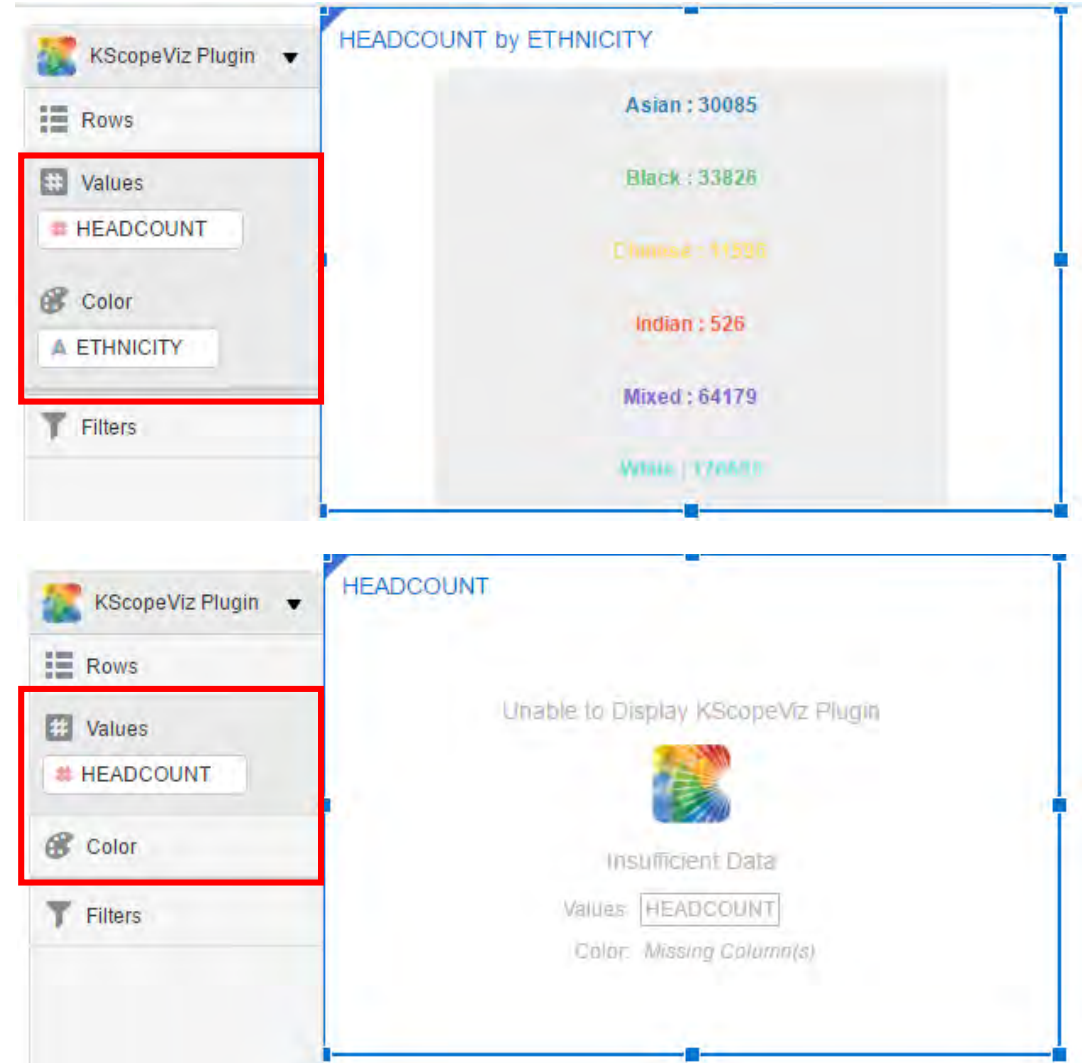
The “row” priority has been made higher than “color”. This is so that data items will by default be placed in the “color” section rather than “row”

In the “color” section we must have 1 data item but no more than 1

We have disabled the “size” section altogether

Data-Model Handler The Result


- After making the modifications to our data-model handler we observe the following:
 - Attributes are placed in to “Color” instead of “Rows” by default
 - The visualization only runs if you have at least one value in “Color”
 - You cannot place more than one metric in “Colour”



PEAK indicators

Data-Model Handler messages.js

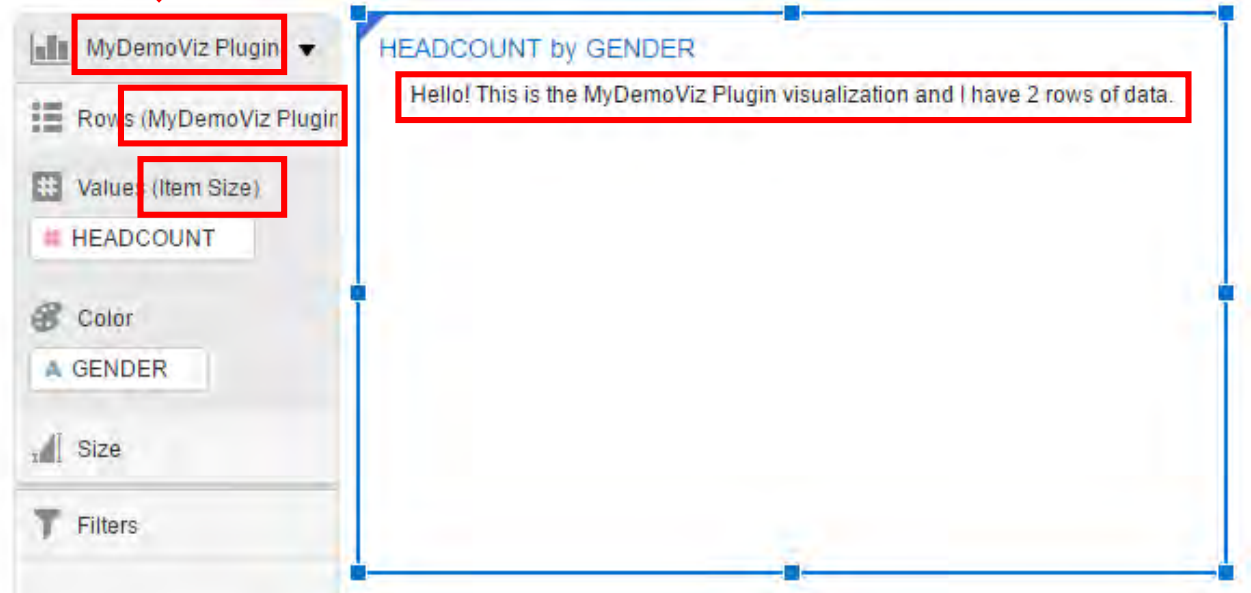
- You can also override certain text headings in the “**messages.js**” file
- This file is located in the **/nls/root** subfolder



OS (C:) > Temp > OracleDV-Custom-Plugins > src > customviz > com-company-myDemoViz > nls > root

Name	Date modified	Type	Size
messages.js	24/06/2017 13:40	JavaScript File	1 KB

```
define({  
  "MYDEMOVIZ_DISPLAY_NAME": "MyDemoViz Plugin",  
  "MYDEMOVIZ_SHORT_DISPLAY_NAME": "MyDemoViz Plugin",  
  "MYDEMOVIZ_CATEGORY": "MyDemoViz Plugin",  
  "MYDEMOVIZ_ROW_LABEL": "MyDemoViz Plugin Row",  
  "MYDEMOVIZ_ITEM_SIZE": "Item Size",  
  "TEXT_MESSAGE": "Hello! This is the {0} visualization and I have {1} rows of data."  
});
```

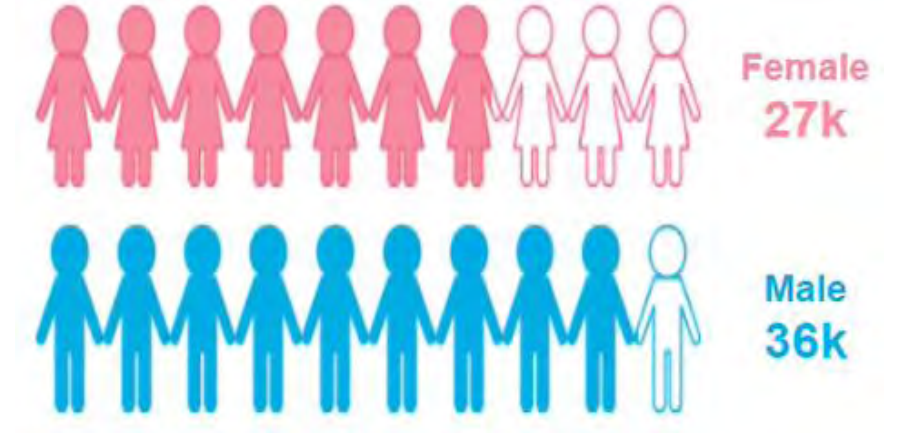


Oracle Data Visualization SDK In Summary

PEAK indicators

In Summary Oracle Data Viz SDK

- SDK provides excellent capabilities for building your own custom visualizations
- Your plugins can be fully integrated into Data Viz
- Template plugin gives you a good head start, your focus can now be on designing and the visuals and being creative with the HTML
- If you build any of your own plugins then contact Oracle and get them uploaded to the public store



PEAK indicators

Further Information

please contact us or visit our web-site:



www.peakindicators.com



enquiries@peakindicators.com



+44 (0) 124 638 9000

