

PUBLIC

# **SAP ANALYTICS CLOUD, ANALYTICS DESIGNER, IN ACTION**

## **ANA362**

Exercises / Solutions

Marouene Ferchichi, Jie Deng, Christina Mast - SAP SE

## TABLE OF CONTENTS

BEFORE YOU START.....	3
ANA362 EXERCISE 1 .....	4
Overview .....	4
Part 1: Add Flow Panel, Slider, Text to the Canvas & Create a Script Variable .....	5
Part 2: Add Charts, Table & Script to Widgets .....	18
Part 3: Changing Theme .....	39
Part 4: Download SAC app from Apple Store and run the application .....	45
Part 5: Embedding into a HTML page.....	50
Summary.....	63

# BEFORE YOU START

Due to time constraints during the Hands-on session, it is recommended that you first take a look at the different exercises and then decide which ones you want to work through first.

## **Exercise 1: ANA362\_Exercise\_1**

Estimated Duration: **[90] minutes**

With this exercise you are going to create a simple responsive analytical application and embed it into a HTML page and run it with your mobile phone.

---

## **Exercise 2: ANA362\_Exercise\_2**

Estimated Duration: **[45] minutes**

With this exercise you will be able to explore and use the Generic Template.

With this template you will have the possibility to select and modify your Data Set using the available functionalities.

---

## **Exercise 3: AIN362\_Exercise\_3**

Estimated Duration: **[45] minutes**

Here you are going to create a simple feedback to ask user to enter the rating of a web page and then calculate the average score of this web page.

You are going to leverage the responsive capability of analytics designer and integration with SAC planning capability to implement this scenario.

---

All the exercises are independent from each other. You can choose the exercise to work on based on your knowledge and interest.

## **Connect to Sap Analytics Cloud**

you can directly type the following address in your **Google Chrome**:

<https://ANA362.eu10.hcs.cloud.sap>

# ANA362 EXERCISE 1

## Overview

Estimated time: [60] minutes

### Objective

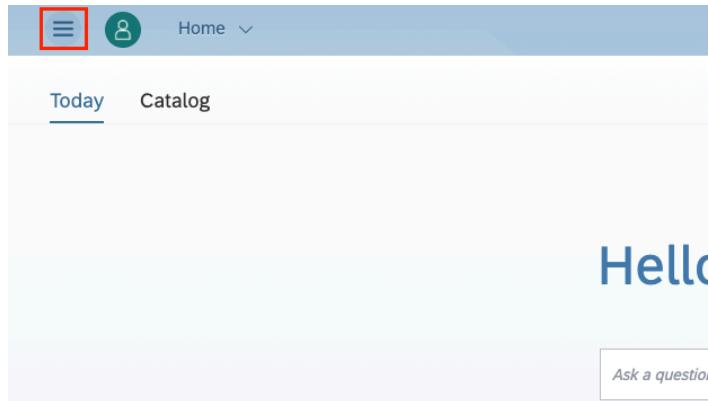
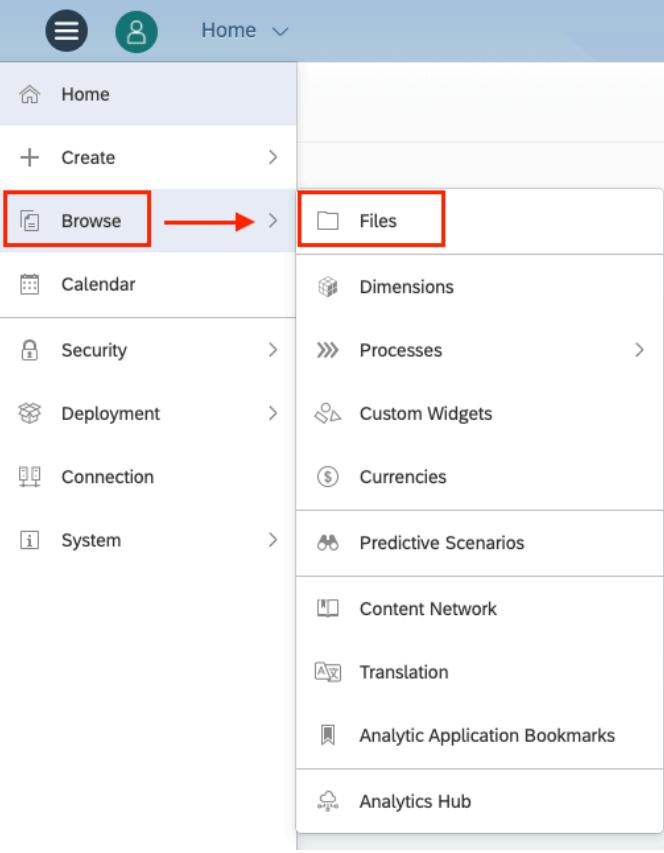
With this exercise you will be able to learn the basic workflow of creating a simple responsive analytical application with SAC Analytics Designer.

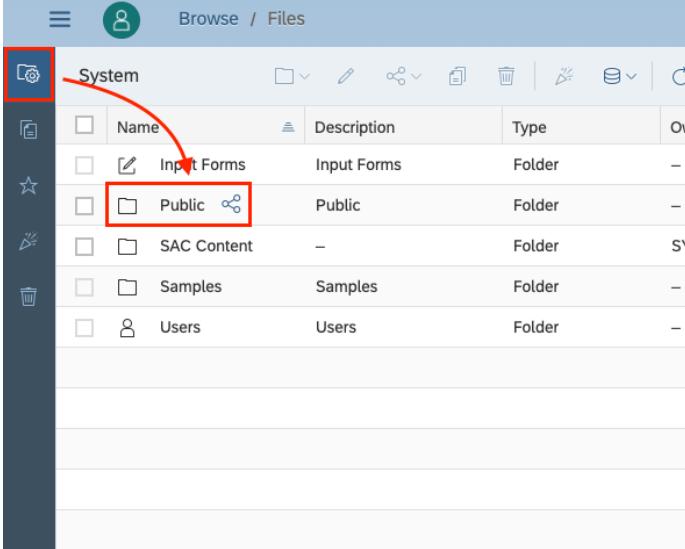
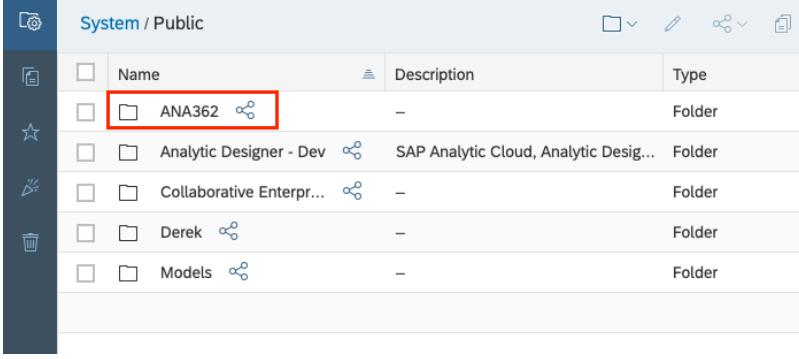
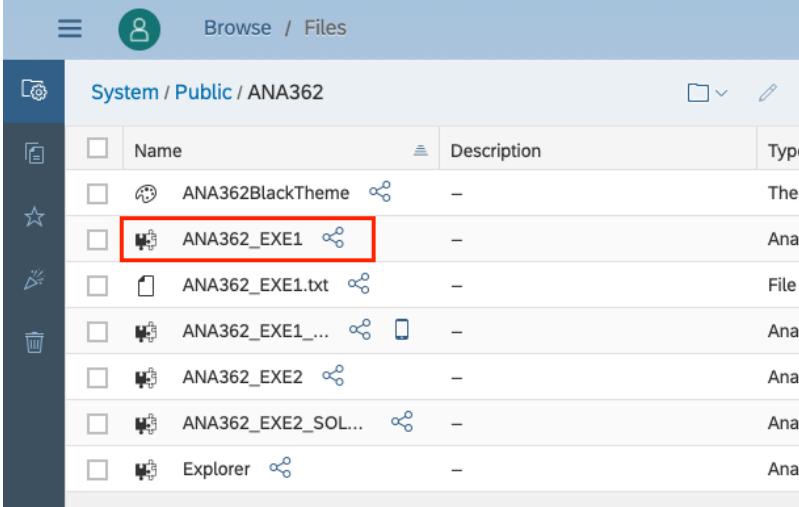
### Exercise Description

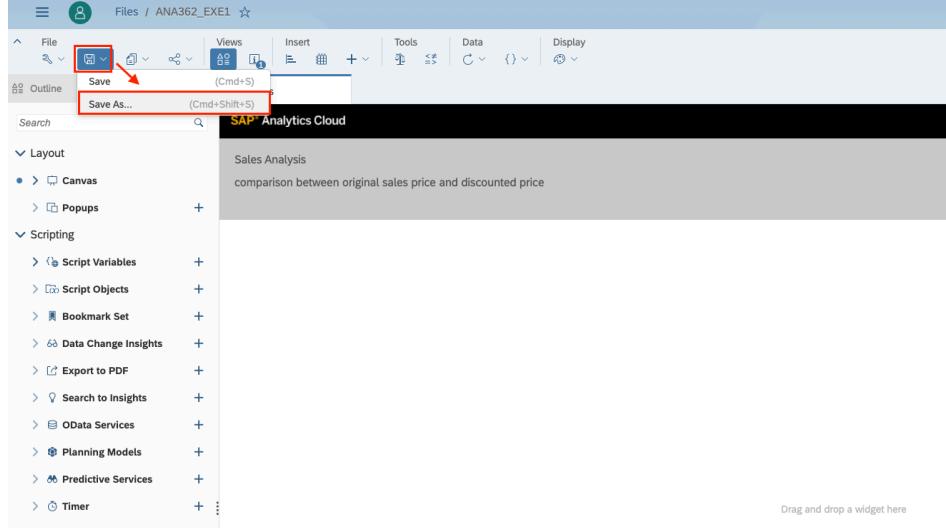
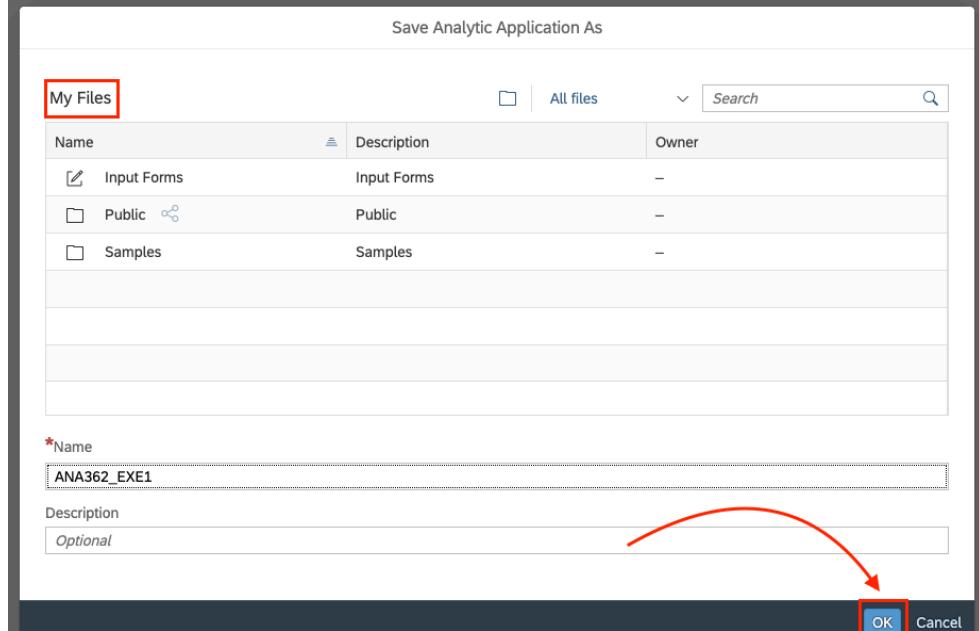
**You are going to execute the following steps within this exercise:**

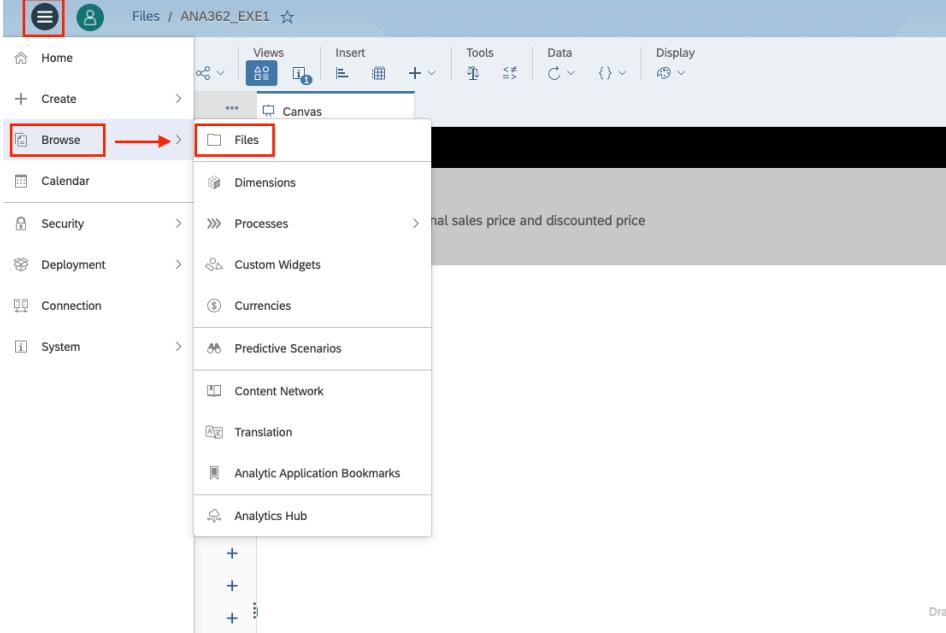
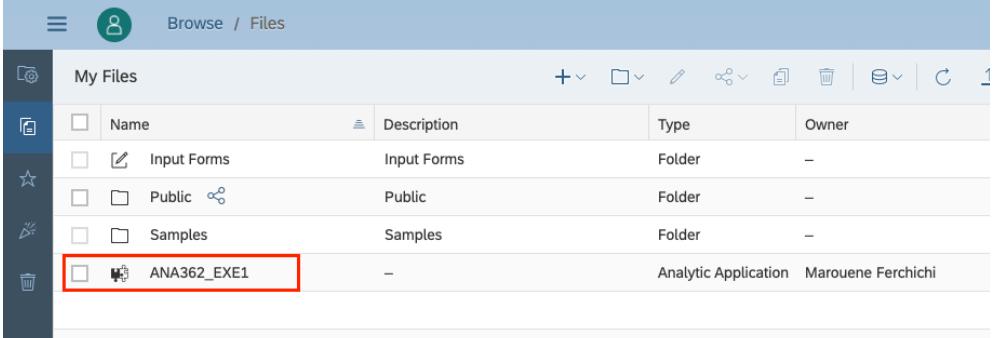
- Add flow panel, charts, table, slider, click event to canvas, configure above objects
- Define Global Script Variable
- Use Global Script Variable for calculated measures
- Set the application to mobile enabled
- Create a theme ANA362\_XXBLACKTHEME, Assign application to theme
- Download SAC iOS Mobile APP and run the application
- Embedding: Add windows post message and post message received, adjust html, and run the application within HTML5

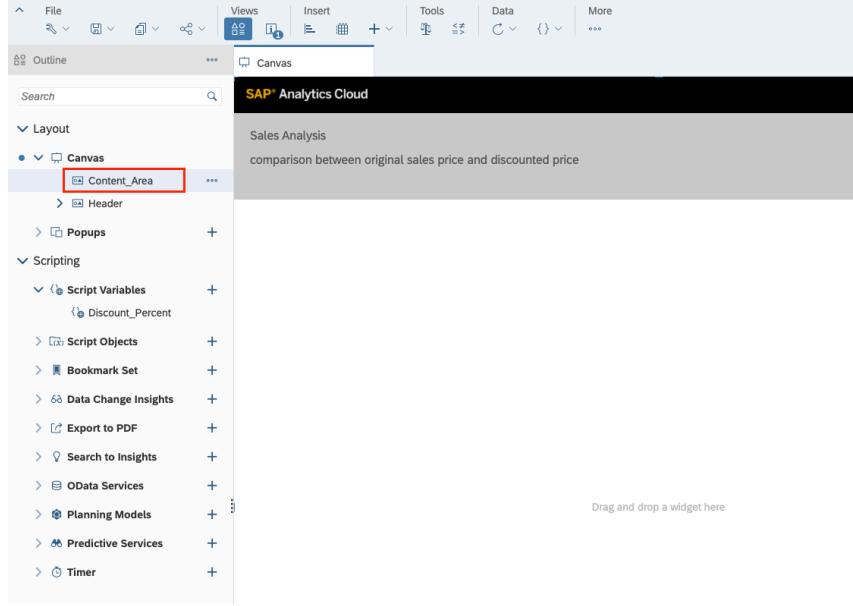
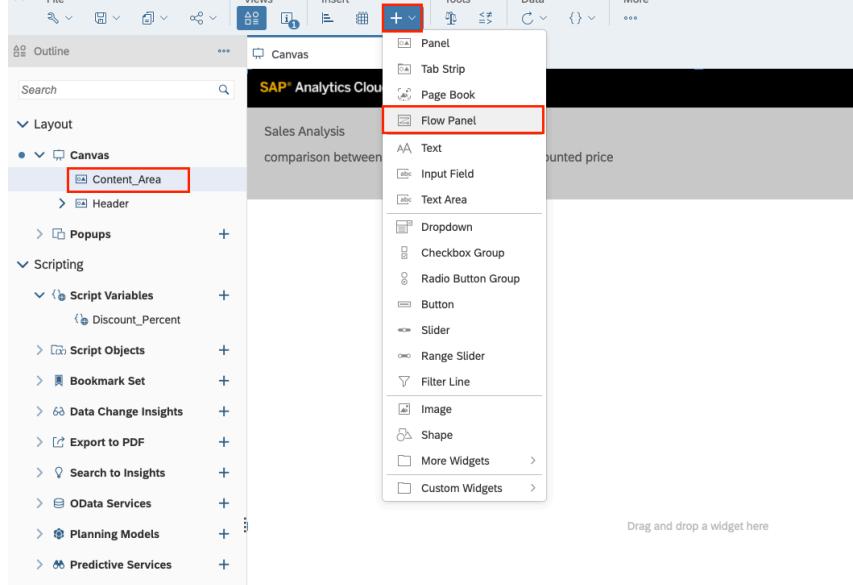
## Part 1: Add Flow Panel, Slider, Text to the Canvas & Create a Script Variable

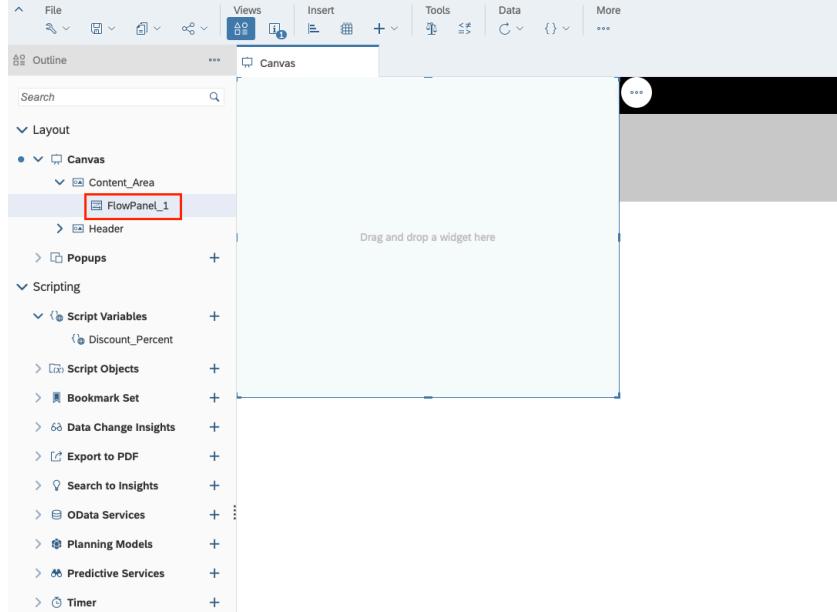
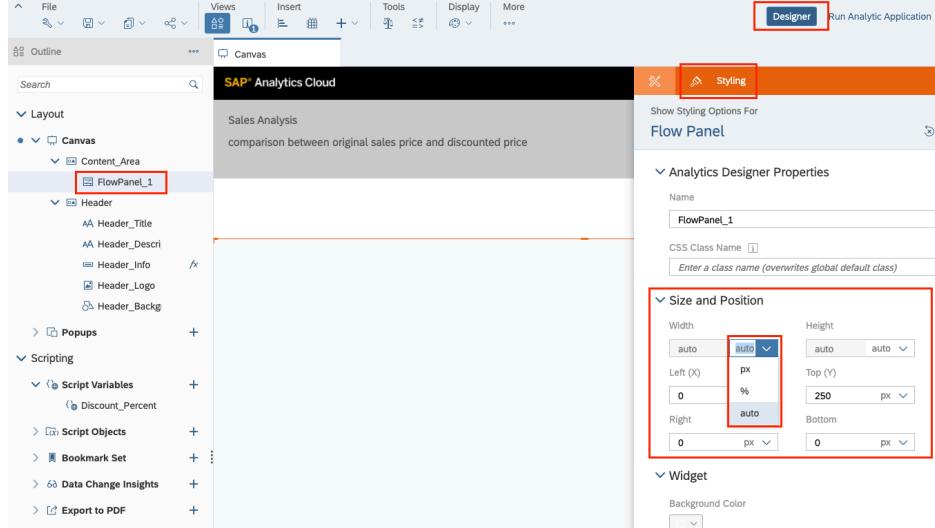
Explanation	Screenshot
1. Click on the <b>Main Menu</b> icon	 <p>The screenshot shows the main menu interface. The top navigation bar includes a main menu icon (three horizontal lines), a user profile icon, and the text "Home". Below the navigation bar, there are two tabs: "Today" (underlined) and "Catalog". A large blue banner with the word "Hello" is visible. At the bottom right, there is a button labeled "Ask a question".</p>
2. Select <b>Browse</b> and then <b>Files</b> to be able to select your analytical applications.	 <p>The screenshot shows the main menu interface with the sidebar expanded. The "Create" section has a red box around the "Browse" option. An arrow points from "Browse" to the "Files" option, which is also highlighted with a red box. The sidebar also lists "Home", "Calendar", "Security", "Deployment", "Connection", and "System". The main content area on the right lists "Dimensions", "Processes", "Custom Widgets", "Currencies", "Predictive Scenarios", "Content Network", "Translation", "Analytic Application Bookmarks", and "Analytics Hub".</p>

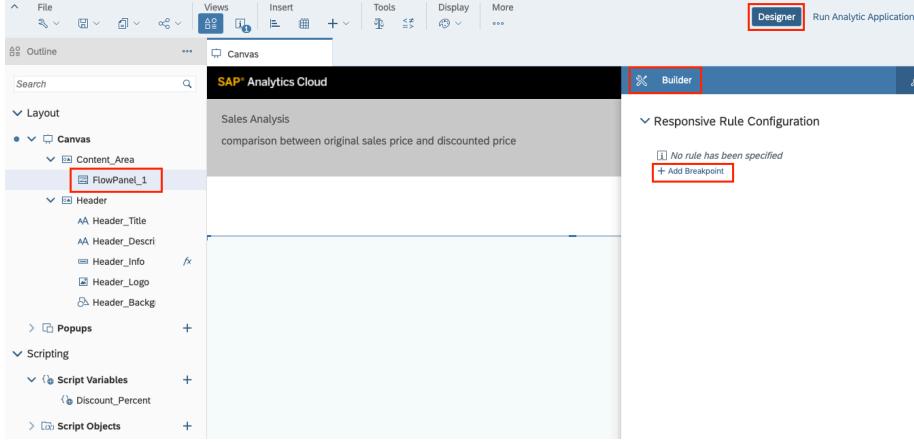
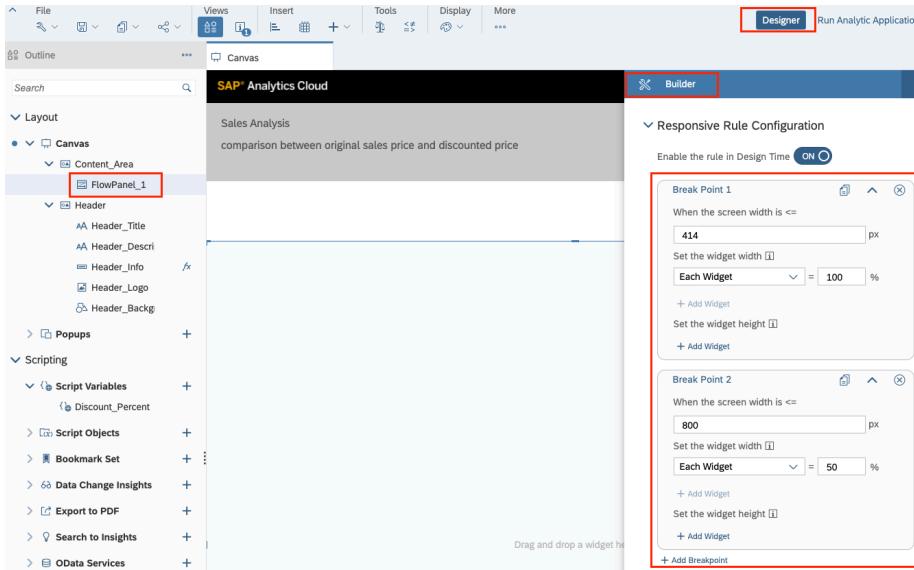
Explanation	Screenshot																								
<p>3. Select the icon <b>System</b>, and then open the folder <b>Public</b>.</p>	 <p>The screenshot shows the SAP Fiori Launchpad with the 'System' icon highlighted in red. A red arrow points from the 'System' icon to the 'Public' folder in the list below. The 'Public' folder is also highlighted with a red box.</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Type</th> </tr> </thead> <tbody> <tr> <td>Input Forms</td> <td>Input Forms</td> <td>Folder</td> </tr> <tr> <td>Public</td> <td>Public</td> <td>Folder</td> </tr> <tr> <td>SAC Content</td> <td>—</td> <td>Folder</td> </tr> <tr> <td>Samples</td> <td>Samples</td> <td>Folder</td> </tr> <tr> <td>Users</td> <td>Users</td> <td>Folder</td> </tr> </tbody> </table>	Name	Description	Type	Input Forms	Input Forms	Folder	Public	Public	Folder	SAC Content	—	Folder	Samples	Samples	Folder	Users	Users	Folder						
Name	Description	Type																							
Input Forms	Input Forms	Folder																							
Public	Public	Folder																							
SAC Content	—	Folder																							
Samples	Samples	Folder																							
Users	Users	Folder																							
<p>4. Navigate to the folder <b>ANA362</b></p>	 <p>The screenshot shows the SAP Fiori Launchpad with the 'ANA362' folder highlighted in red.</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Type</th> </tr> </thead> <tbody> <tr> <td>ANA362</td> <td>—</td> <td>Folder</td> </tr> <tr> <td>Analytic Designer - Dev</td> <td>SAP Analytic Cloud, Analytic Desig...</td> <td>Folder</td> </tr> <tr> <td>Collaborative Enterpr...</td> <td>—</td> <td>Folder</td> </tr> <tr> <td>Derek</td> <td>—</td> <td>Folder</td> </tr> <tr> <td>Models</td> <td>—</td> <td>Folder</td> </tr> </tbody> </table>	Name	Description	Type	ANA362	—	Folder	Analytic Designer - Dev	SAP Analytic Cloud, Analytic Desig...	Folder	Collaborative Enterpr...	—	Folder	Derek	—	Folder	Models	—	Folder						
Name	Description	Type																							
ANA362	—	Folder																							
Analytic Designer - Dev	SAP Analytic Cloud, Analytic Desig...	Folder																							
Collaborative Enterpr...	—	Folder																							
Derek	—	Folder																							
Models	—	Folder																							
<p>5. Click on the exercise <b>ANA362_EXE1</b> to open it and then you will save it with <b>My Files</b> folder.</p>	 <p>The screenshot shows the SAP Fiori Launchpad with the 'ANA362_EXE1' file highlighted in red.</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Description</th> <th>Type</th> </tr> </thead> <tbody> <tr> <td>ANA362BlackTheme</td> <td>—</td> <td>Theme</td> </tr> <tr> <td>ANA362_EXE1</td> <td>—</td> <td>Anal...</td> </tr> <tr> <td>ANA362_EXE1.txt</td> <td>—</td> <td>File</td> </tr> <tr> <td>ANA362_EXE1_...</td> <td>—</td> <td>Anal...</td> </tr> <tr> <td>ANA362_EXE2</td> <td>—</td> <td>Anal...</td> </tr> <tr> <td>ANA362_EXE2_SOL...</td> <td>—</td> <td>Anal...</td> </tr> <tr> <td>Explorer</td> <td>—</td> <td>Anal...</td> </tr> </tbody> </table>	Name	Description	Type	ANA362BlackTheme	—	Theme	ANA362_EXE1	—	Anal...	ANA362_EXE1.txt	—	File	ANA362_EXE1_...	—	Anal...	ANA362_EXE2	—	Anal...	ANA362_EXE2_SOL...	—	Anal...	Explorer	—	Anal...
Name	Description	Type																							
ANA362BlackTheme	—	Theme																							
ANA362_EXE1	—	Anal...																							
ANA362_EXE1.txt	—	File																							
ANA362_EXE1_...	—	Anal...																							
ANA362_EXE2	—	Anal...																							
ANA362_EXE2_SOL...	—	Anal...																							
Explorer	—	Anal...																							

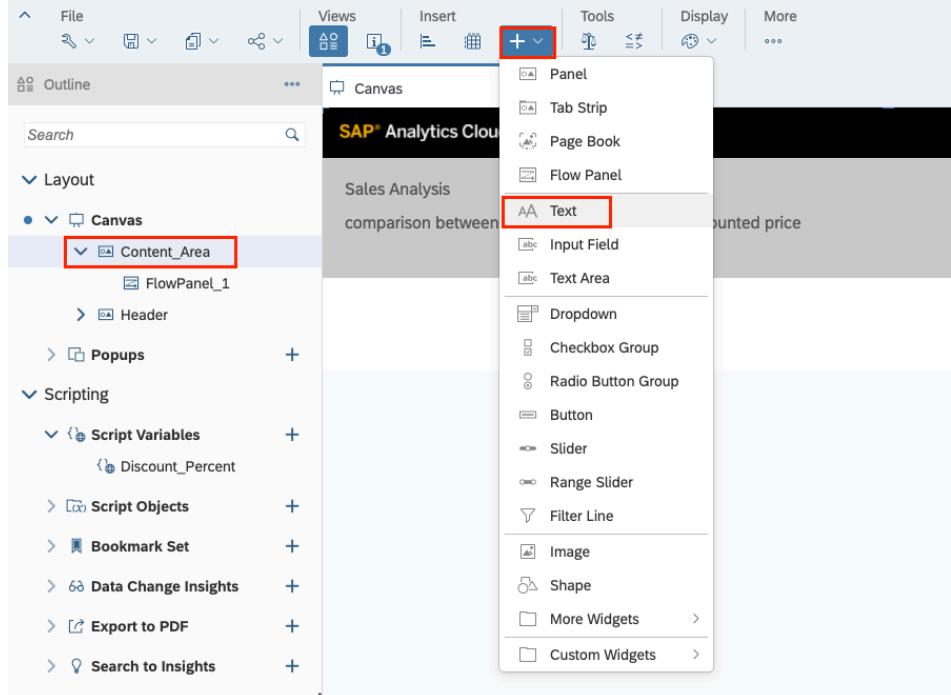
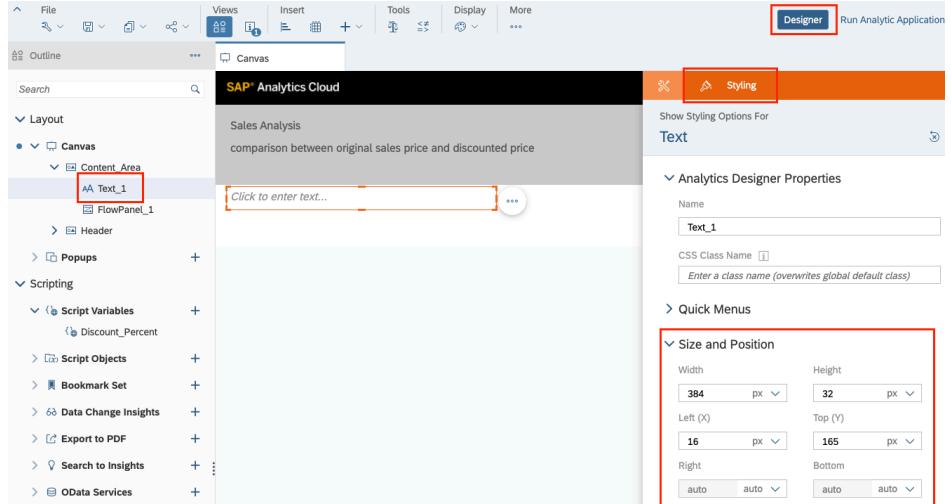
Explanation	Screenshot
<p>6. Before you start the different steps of the exercise, first of all <b>YOU SHOULD</b> save the application under <b>My Files</b> and rename it so you can work on your own application.</p> <p>Click on the <b>Save</b> icon → <b>Save As</b></p>	
<p>7. Be sure that you are under the folder <b>My Files</b>.</p> <p>Press <b>OK</b> and the application will be saved under your private folder <b>My Files</b></p>	

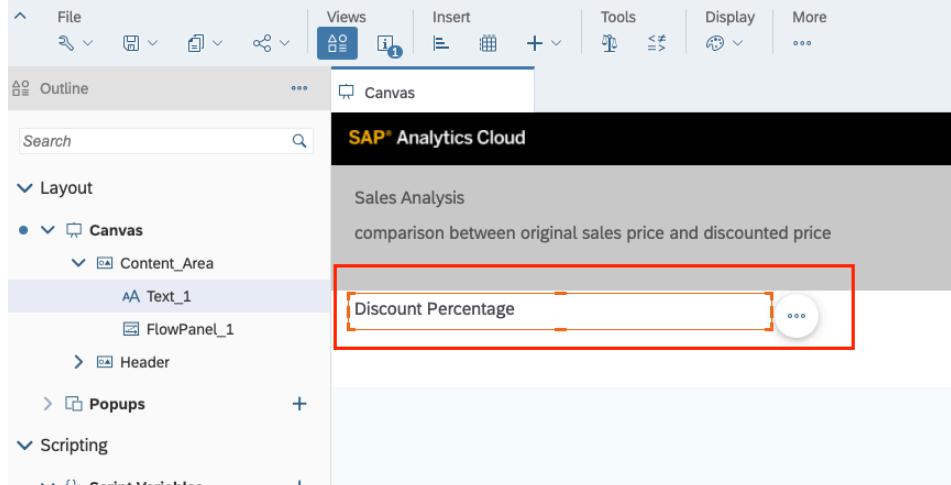
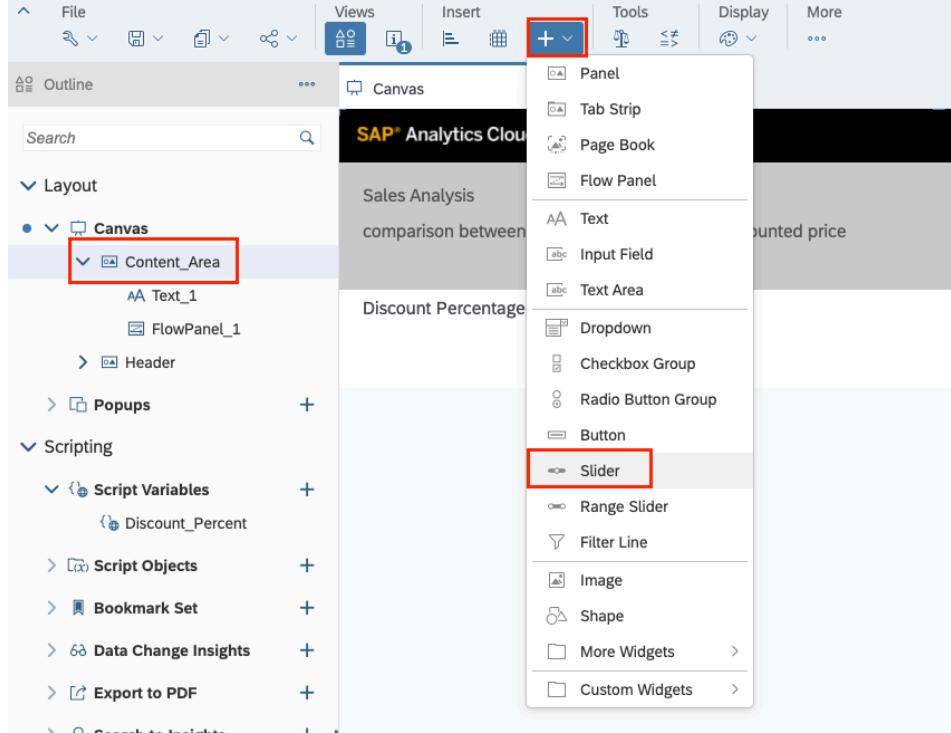
Explanation	Screenshot
<p>8. Now you can navigate back to <b>My Files</b> again in order to see your application.</p>	 <p>The screenshot shows the SAP Fiori Launchpad interface. On the left, there's a navigation bar with icons for Home, Create, Browse, Calendar, Security, Deployment, Connection, and System. The 'Browse' icon is highlighted with a red box and has a red arrow pointing to it. On the right, there's a list of items under 'Views': Canvas, Files (which is also highlighted with a red box), Dimensions, Processes, Custom Widgets, Currencies, Predictive Scenarios, Content Network, Translation, Analytic Application Bookmarks, and Analytics Hub. Below this is a section with three '+' buttons and a 'Drag' placeholder.</p>
<p>9. Click on the exercise <b>ANA362_EXE1</b> to open it within <b>My Files</b> folder.</p>	 <p>The screenshot shows the 'My Files' list view. It has a header with columns: Name, Description, Type, and Owner. There are four entries: 'Input Forms' (Folder, Owner -), 'Public' (Folder, Owner -), 'Samples' (Folder, Owner -), and 'ANA362_EXE1' (Analytic Application, Owner: Marouene Ferchichi). The 'ANA362_EXE1' entry is highlighted with a red box.</p>

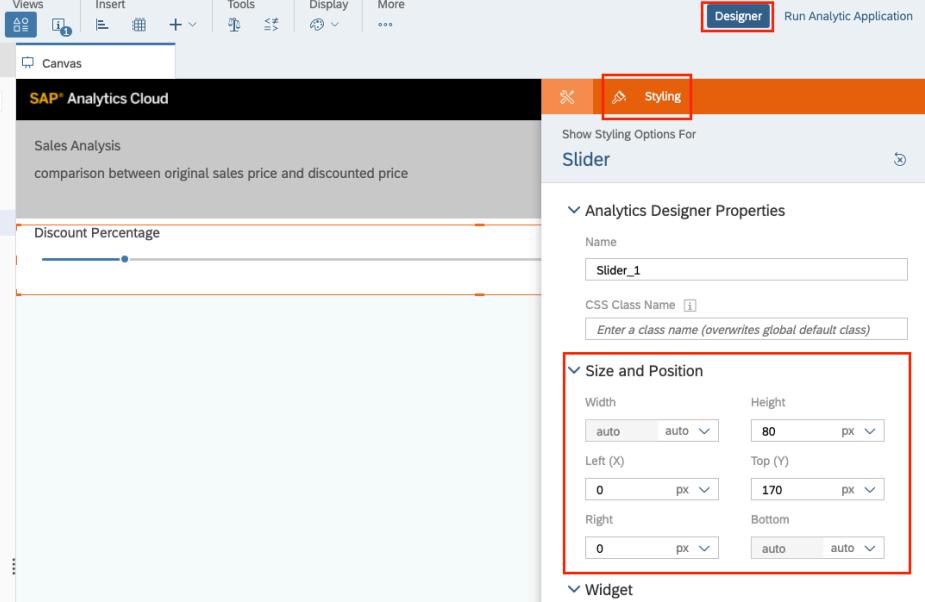
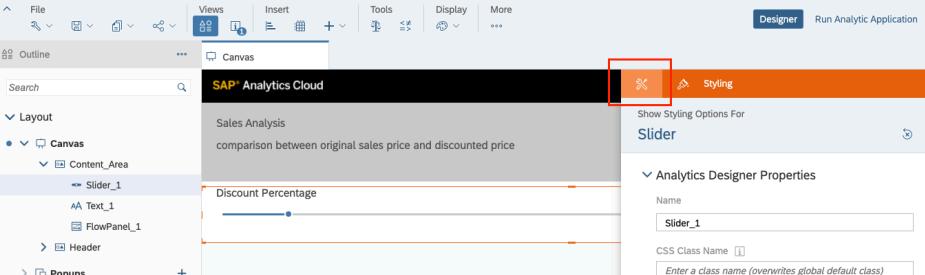
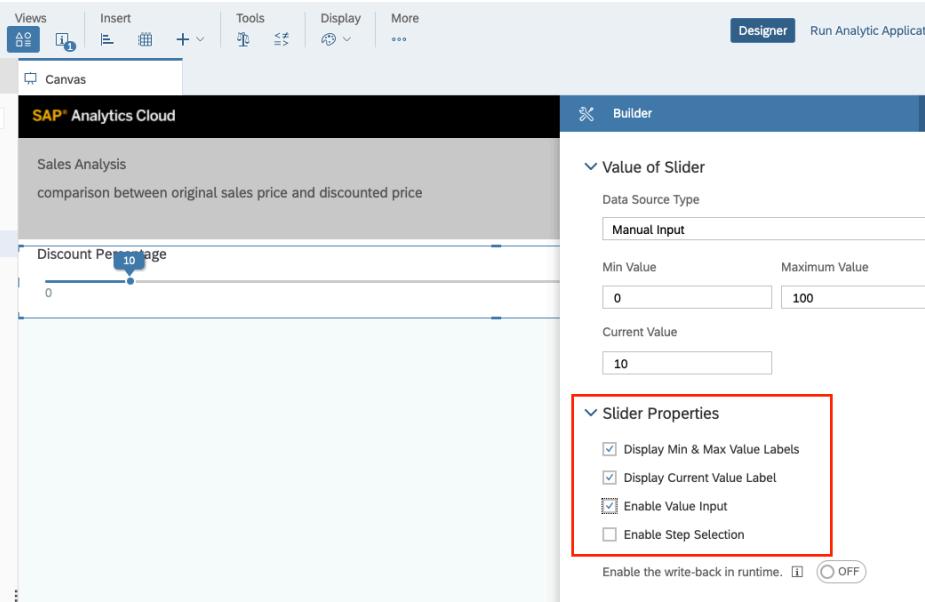
Explanation	Screenshot
<p>10. Now you will add a <b>Flow Panel</b> widget to the canvas under the Content Area. Please switch to the Content Area by clicking on <b>Content_Area</b> from the layout panel on the left.</p>	
<p>11. Move to insert from the toolbar and click on the + sign, then select the widget <b>Flow Panel</b>.</p>	

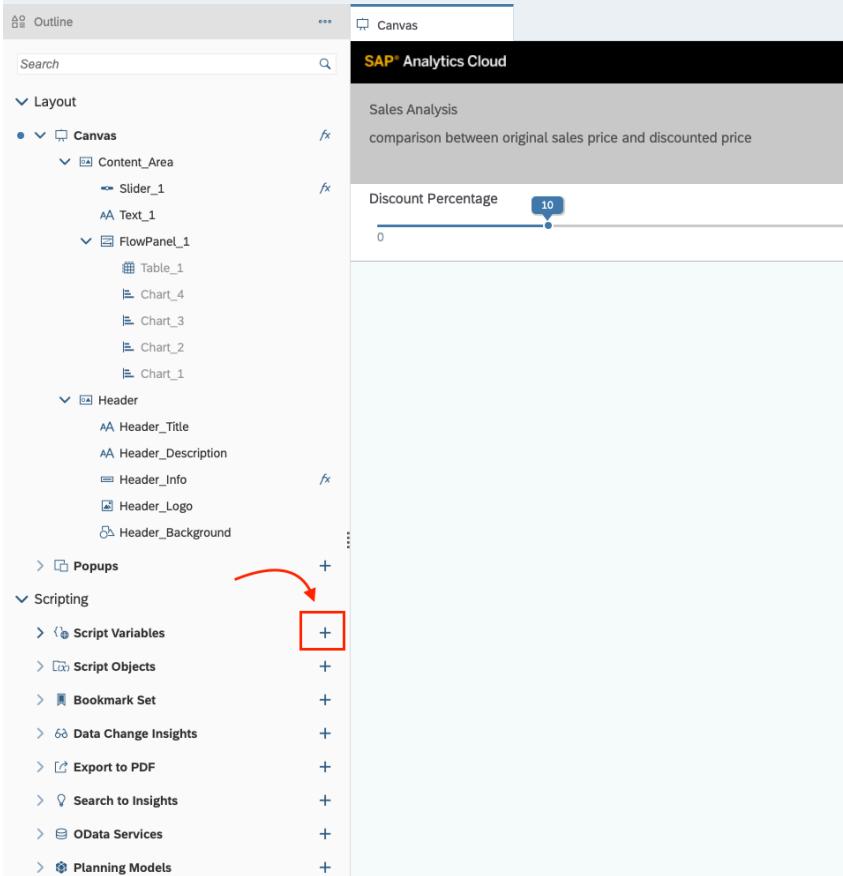
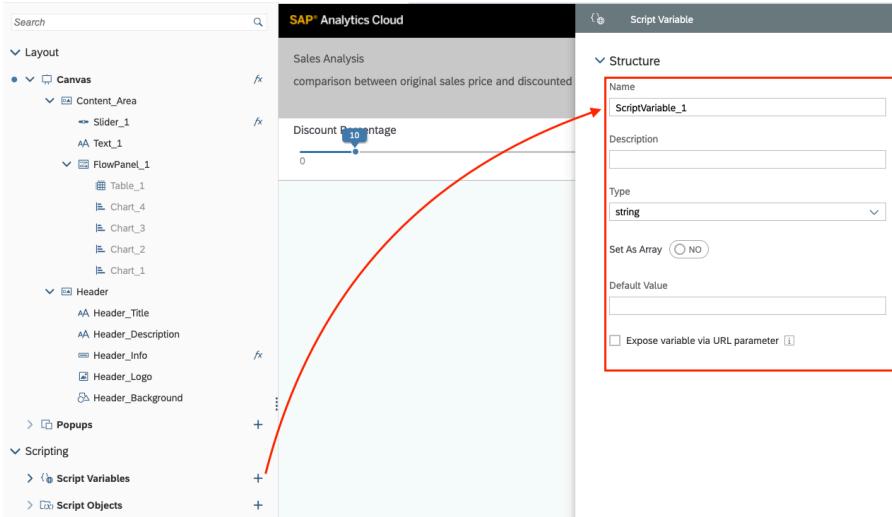
Explanation	Screenshot
<p>12. As you can see that the new widget (<b>Flow Panel</b>) was added as a new component in the Layout under Content Area.</p> <p>Please pay attention, FlowPanel_1 should be under the Content Area.</p>	
<p>13. Define the layout configuration of the <b>Flow Panel</b> by opening the designer panel then to Styling.</p> <p>Change the default configuration to:</p> <p><b>Width: auto</b>  <b>Height: auto</b>  <b>Left: 0 px</b>  <b>Top: 250 px</b>  <b>Right: 0 px</b>  <b>Bottom: 0 px</b></p>	

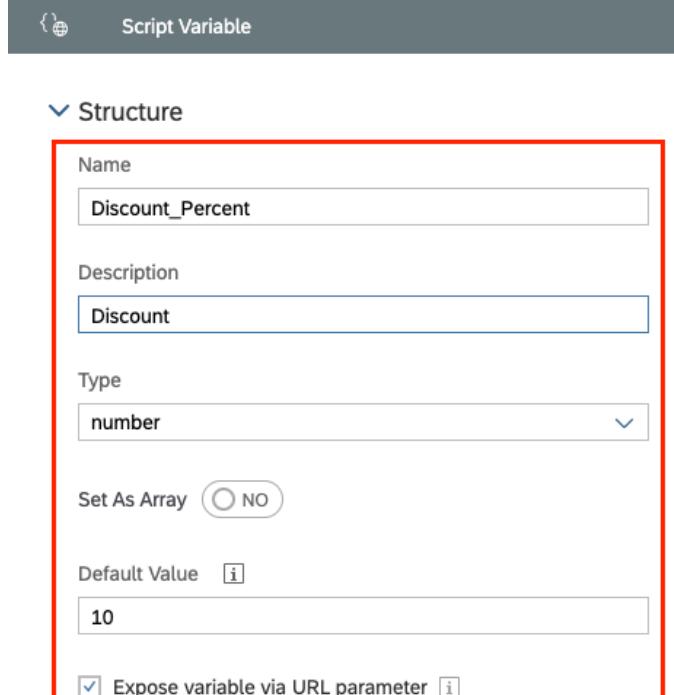
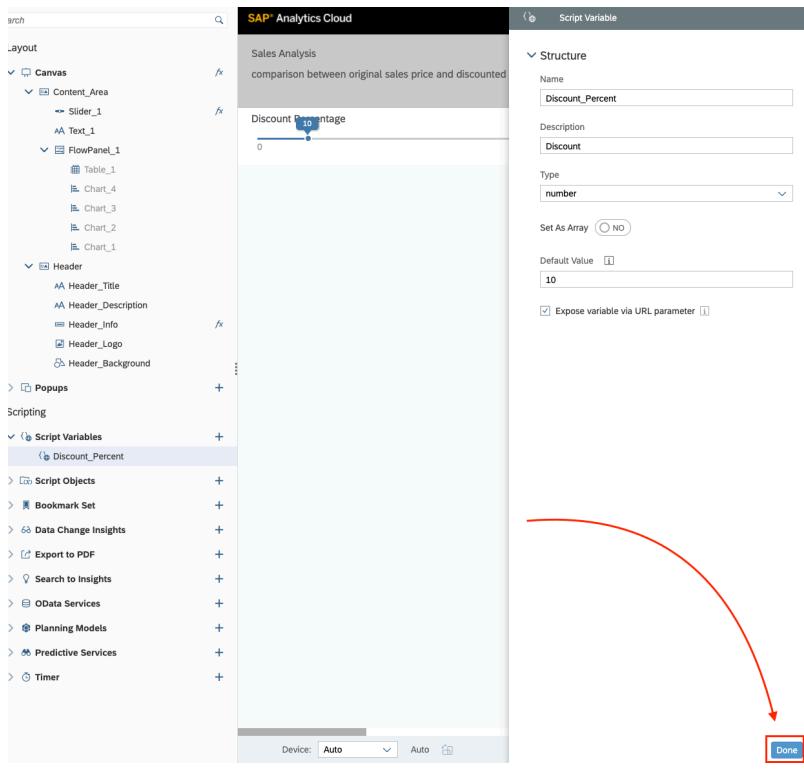
Explanation	Screenshot
<p>14. Define the responsive rule configuration of the <b>Flow Panel</b> by adding 2 Break point in order to set the behave of the widget with the different screen size.</p>	
<p>15. Define the break points as shown in the screenshot:</p> <p><b>Break Point 1:</b> Screen Width: 414 px  Widget width: Each widget = 100%</p> <p><b>Break Point 2:</b> Screen Width: 800 px  Widget width: Each widget = 50%</p>	

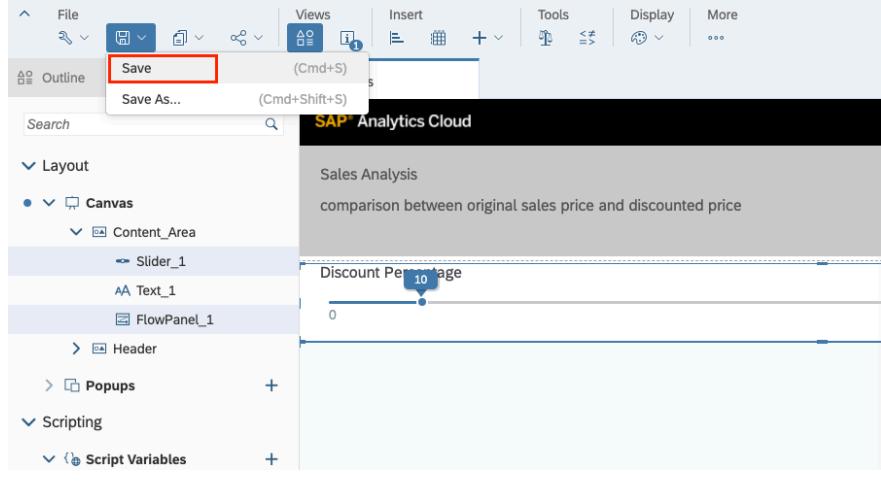
Explanation	Screenshot
<p><b>16. Let's add a Text Widget.</b></p> <p>This widget will be used as a description for the <b>Slider</b> that we will insert later.</p> <p>Click on the (+) sign from the insert toolbar and then select Text.</p> <p>You should select <b>Content_Area</b> from the Outline Panel in order to have the <b>Text Widget</b> under Content Area.</p>	
<p><b>17. Define the layout configuration of the Text widget:</b></p> <p><b>Width:</b> 384 px  <b>Height:</b> 32 px  <b>Left:</b> 16 px  <b>Top:</b> 165 px  <b>Right:</b> auto  <b>Bottom:</b> auto</p>	

Explanation	Screenshot
<p>18. Enter <b>Discount Percentage</b> in the text field.</p>	
<p>19. Now we will add a <b>Slider</b> widget to the Content Area.</p> <p>Click on the plus sign again (+) to insert the <b>Slider</b>.</p>	

Explanation	Screenshot
<p>20. Define the layout configuration of the <b>Slider</b> widget:</p> <p><b>Width:</b> auto  <b>Height:</b> 80 px  <b>Left:</b> 0 px  <b>Top:</b> 170 px  <b>Right:</b> 0 px  <b>Bottom:</b> auto</p>	
<p>21. Toggle the <b>Designer</b> panel to the <b>Builder</b> panel.</p>	
<p>22. Select the following options for the Slider:</p> <ul style="list-style-type: none"> <li><input checked="" type="checkbox"/> Display Min &amp; Max Value Labels</li> <li><input checked="" type="checkbox"/> Display Current Value Label</li> <li><input checked="" type="checkbox"/> Enable Value Input</li> <li><input type="checkbox"/> Enable Step Selection</li> </ul>	

Explanation	Screenshot
<p>23. With the previous step we defined the <b>Slider</b> as widget. Now, we are going to define a script variable to represent the current selected (via slider) discount value.</p> <p>Go to <b>Scripting</b> → <b>Script Variable</b> View. Click on the icon + to add a new <b>Variable</b>.</p>	
<p>24. As next a new Panel will be opened. Let's Configure our <b>Script Variable</b>.</p>	

Explanation	Screenshot
<p>25. Let's enter some configurations:</p> <p>Name: <b>Discount_Percent</b></p> <p>Description: <b>Discount</b></p> <p>Type: <b>number</b> (because the variable should represent numerical values)</p> <p>Default Value: <b>10</b></p> <p><b>Check the: "Expose variable via URL parameter" (This option enables you to pass script variable values via a URL parameter.)</b></p>	 <p>The screenshot shows the SAP Analytics Cloud interface with the 'Script Variable' configuration panel open. The 'Structure' section is highlighted with a red box. It contains the following fields:</p> <ul style="list-style-type: none"> <li>Name: Discount_Percent</li> <li>Description: Discount</li> <li>Type: number</li> <li>Set As Array: NO (radio button)</li> <li>Default Value: 10</li> <li><input checked="" type="checkbox"/> Expose variable via URL parameter</li> </ul>
<p>26. Click the button <b>Done</b> to close the configuration panel.</p>	 <p>The screenshot shows the SAP Analytics Cloud interface with the configuration panel closed. A red arrow points from the 'Done' button at the bottom right of the configuration panel to the 'Done' button at the bottom right of the main interface.</p>

Explanation	Screenshot
27. Click on the <b>floppy disk</b> icon, then select Save from the display options.	 A screenshot of the SAP Analytics Cloud interface. The top navigation bar includes File, Views, Insert, Tools, Display, and More. The File menu has a 'Save' option highlighted with a red box. Below the menu is a search bar and a 'Layout' sidebar on the left. The main canvas area shows a 'Sales Analysis' card with the text 'comparison between original sales price and discounted price'. A slider widget labeled 'Discount Percentage' is set to 10. The layout sidebar lists components like Content_Area, Slider_1, Text_1, FlowPanel_1, Header, and Popups.

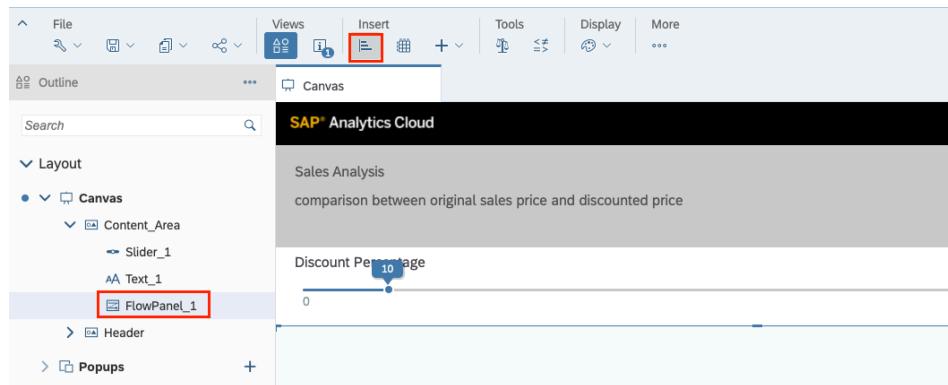
Congratulations! You just finished the part 1 of Exercise 1.

In the **Part 2** of the exercise we are going to add different type of charts and table widget to the Canvas, also writing simple script for widgets

---

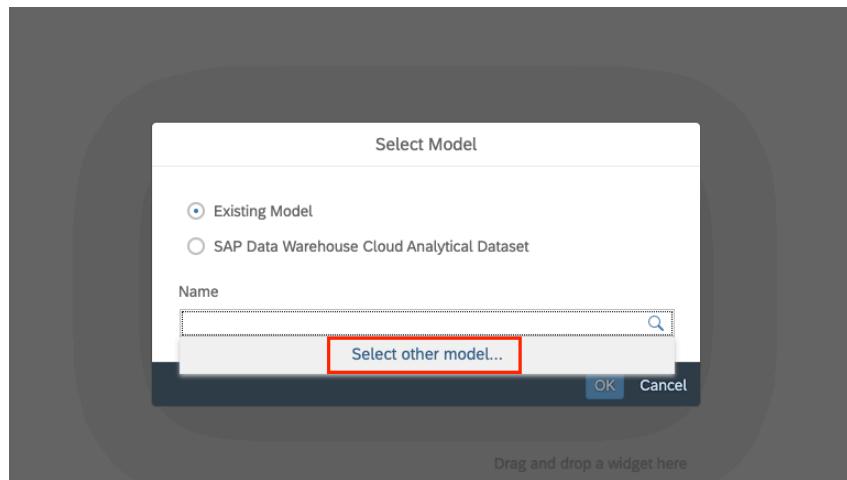
## Part 2: Add Charts, Table & Script to Widgets

- Click on the **Chart icon** to add a **Chart (bar chart)** to the canvas under **FlowPanel\_1** which should be selected from the outline panel.



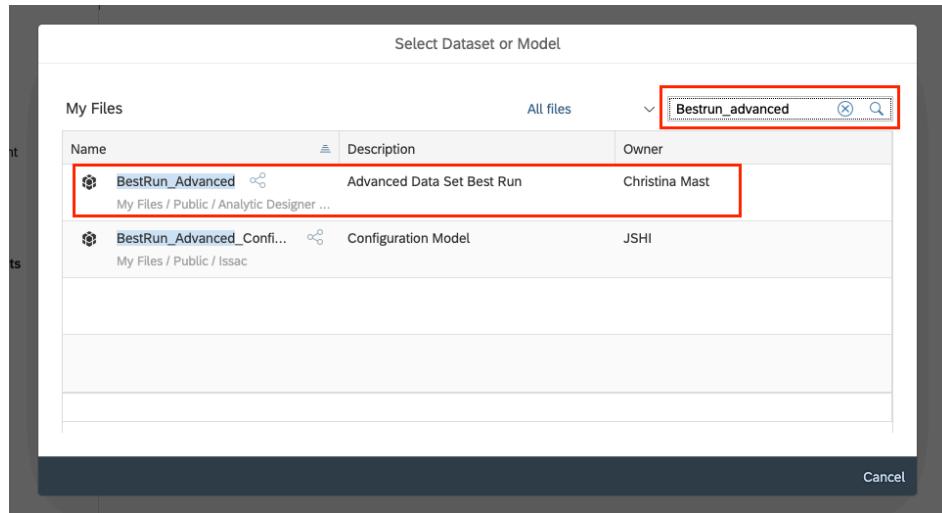
- Now you need to select a data source for the Chart widget. Please use **BestRun\_Advanced** as a data source.

To find the data source click on **Select other model**



- Use the **search bar** on the top right from the Pop up to search for our data source as shown on the screenshot:

After you find the data source **BestRun\_Advanced** Please select it



4. Toggle to the **Builder** panel to add measure(s) and dimension(s)

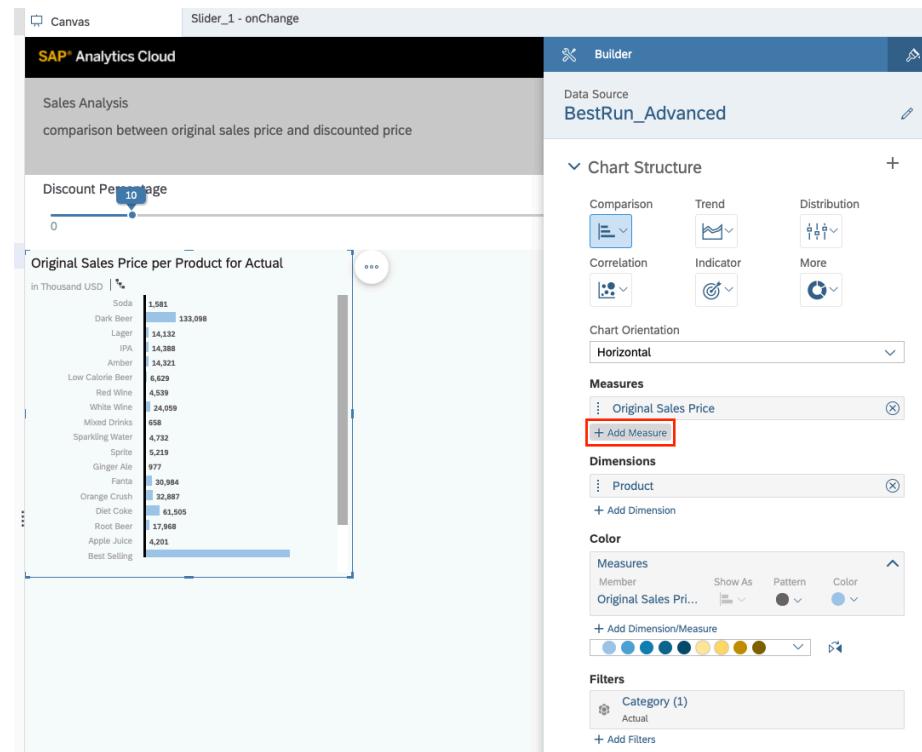
5. Select **Product** as a dimension

6. Select **Original Sales Price** as a measure via Add Measure

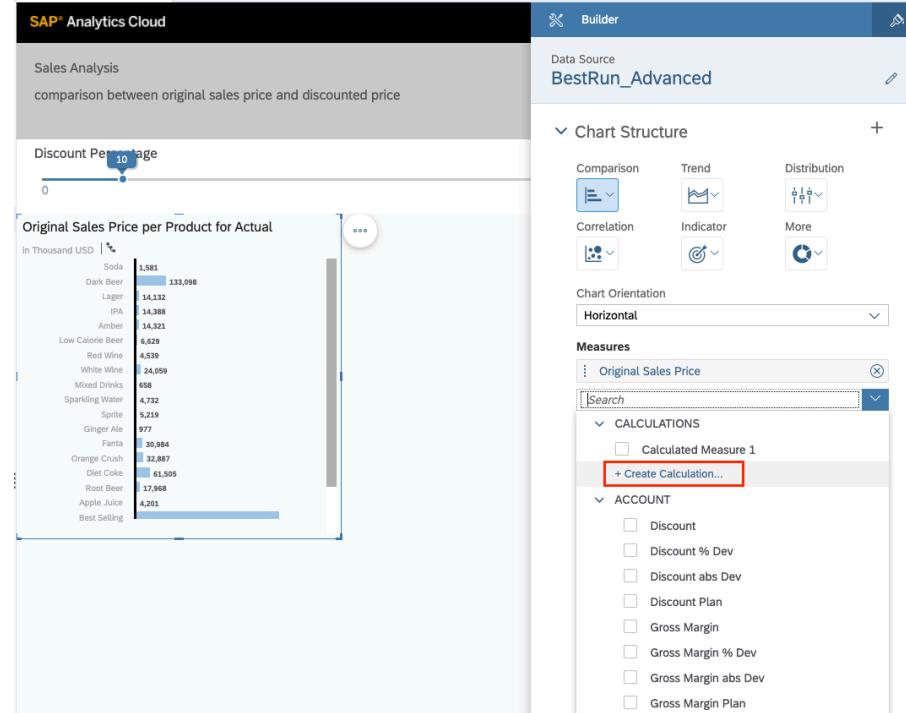
Measure	Value
Original Sales Price	133,098
Dark Beer	14,132
Lager	14,388
IPA	14,321
Amber	6,429
Low Calorie Beer	4,539
Red Wine	24,059
White Wine	658
Mixed Drinks	4,732
Sparkling Water	5,219
Sprite	977
Ginger Ale	30,984
Fanta	32,887
Orange Crush	61,505
Diet Coke	17,968
Root Beer	4,201
Apple Juice	
Best Selling	

7. Let's add a **Calculated Measure** based on the script variable which we already defined in the previous part of the exercise.

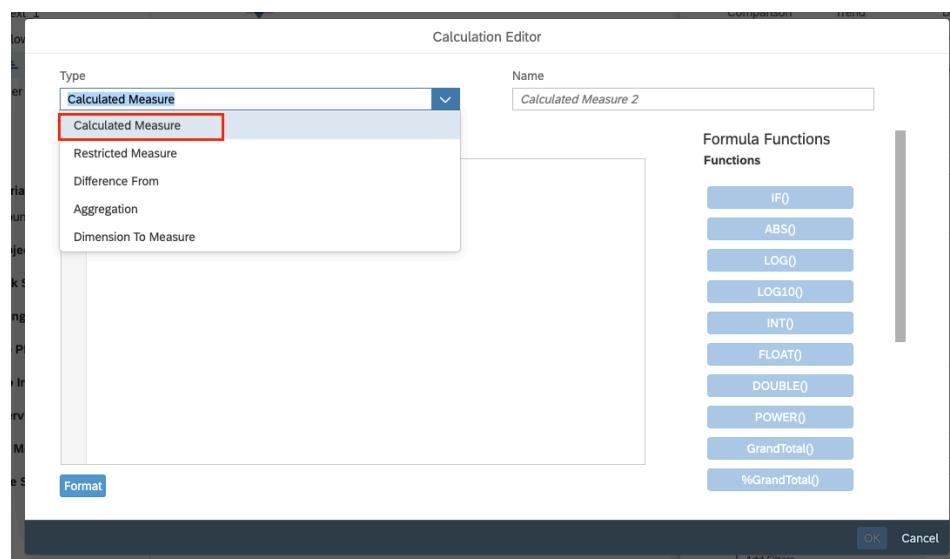
**Click add Measure**



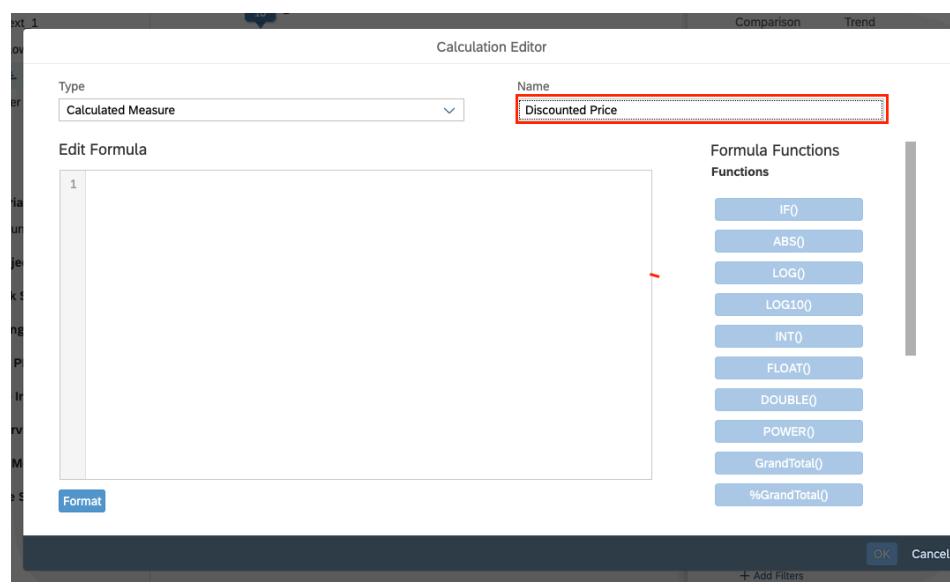
8. Click on **Create Calculation ...**



9. Select **Calculated Measure** as Type.



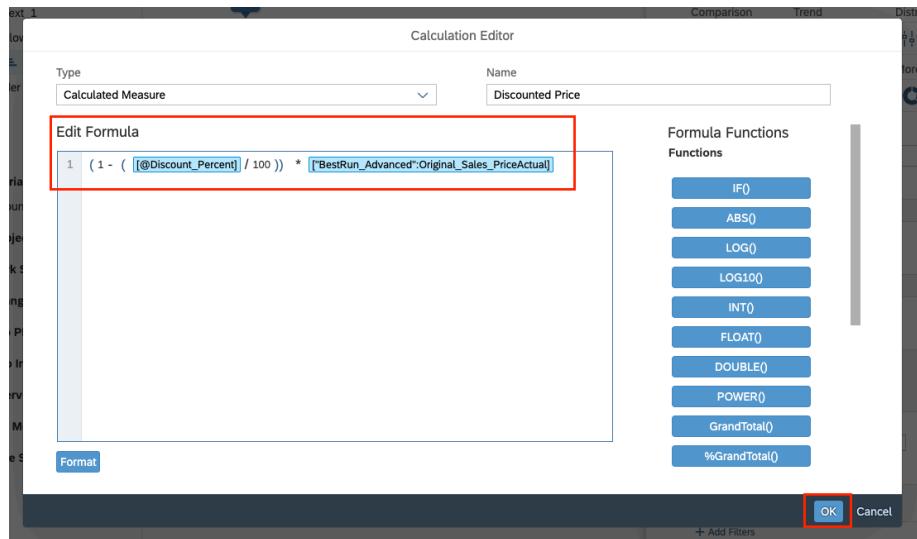
10. Enter the name of the calculated measure:  
**Discounted Price**



11. Edit the formula and enter:

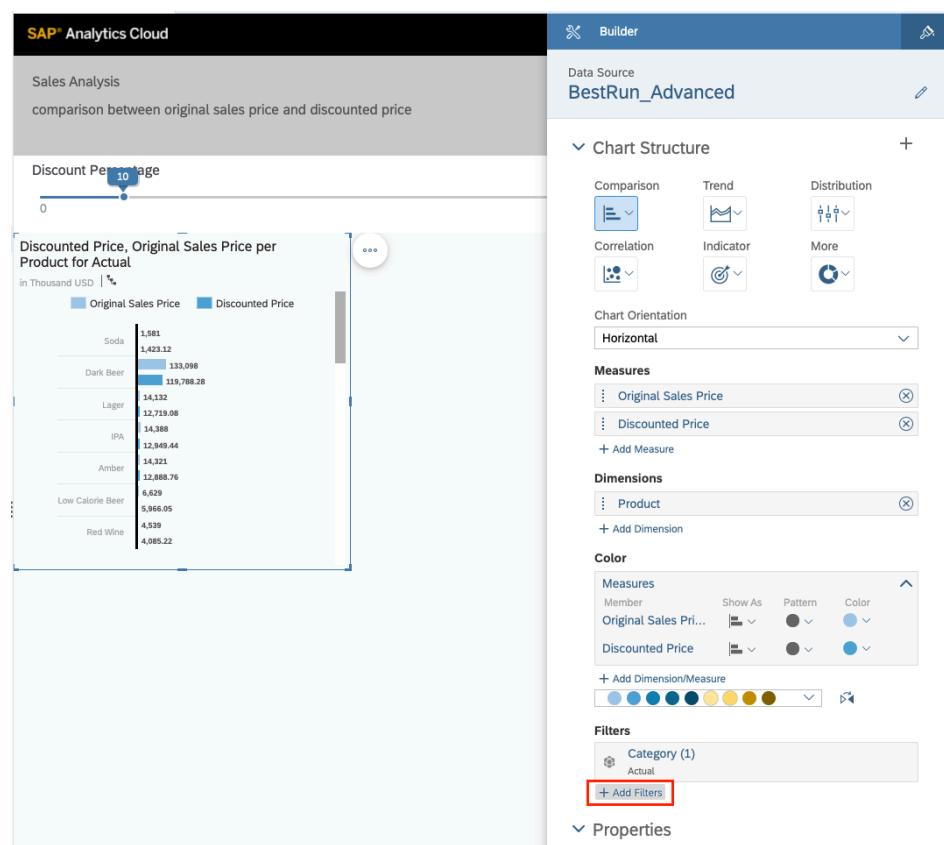
$(1 - (@Discount_Percent)/100) * ["BestRun_Advanced":Original_Sales_PriceActual]$

Press **OK** to confirm and to close the Calculation editor as well.

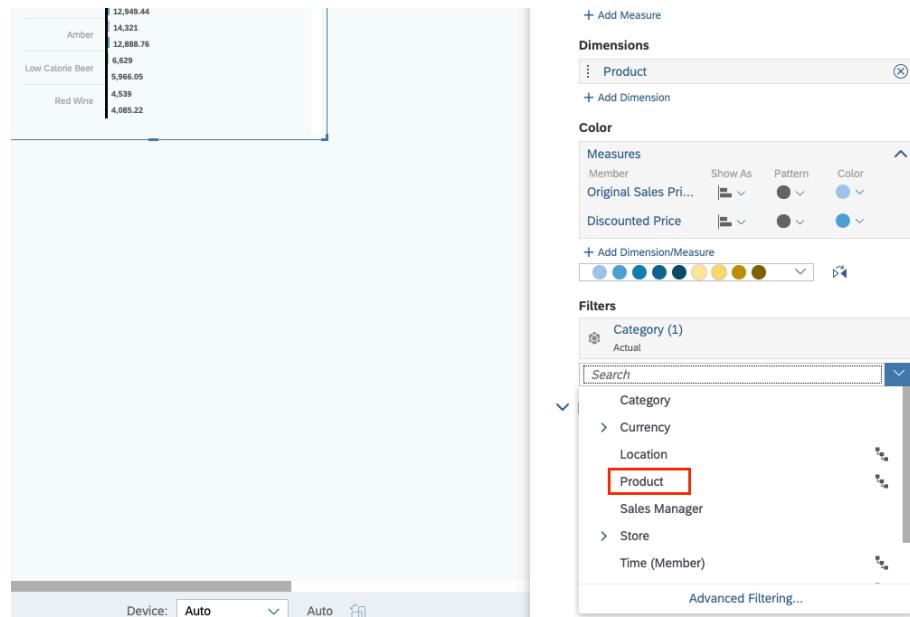


12. After we added 2 measures and 1 dimension, let's add a **filter** in order to show only some specific set of data in the Chart. In that case we will filter on the **Product**:

Click on **Add Filter**



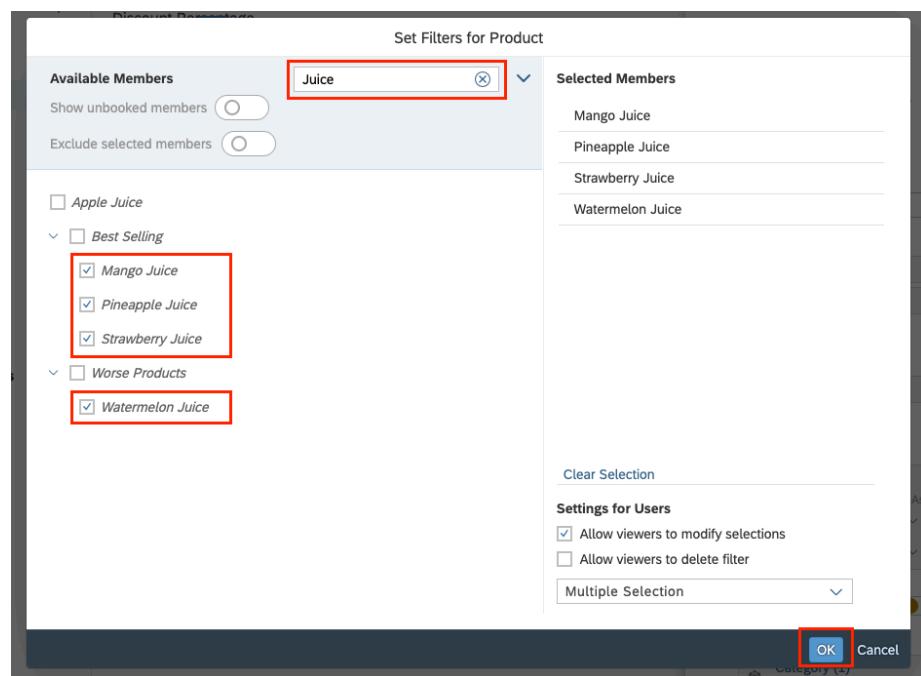
13. Click on **Product**. we want to filter on the Product dimension.



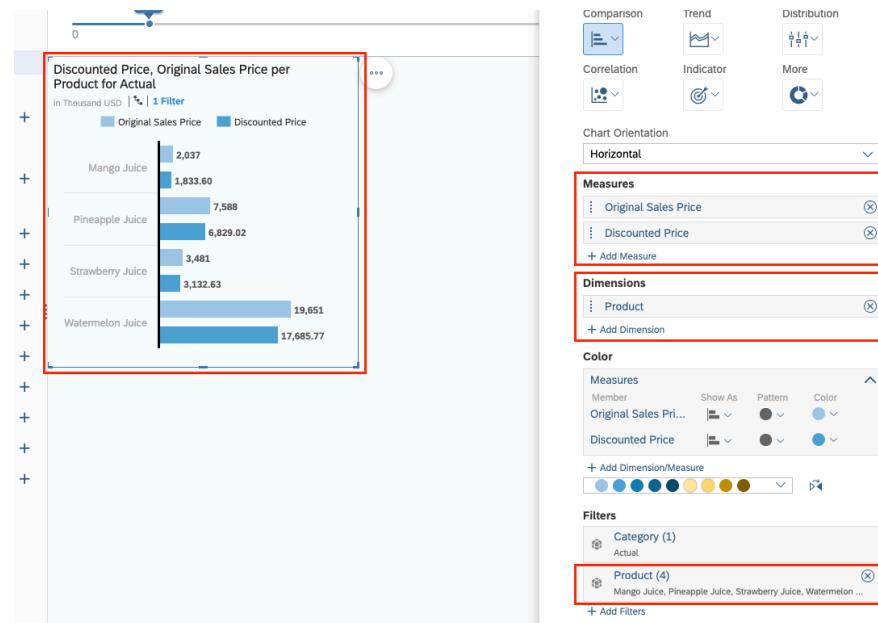
14. Search for **Juice**.

Select **Mango Juice**, **Pineapple Juice**, **Strawberry Juice** and **Watermelon Juice**.

15. Click **OK** to confirm.

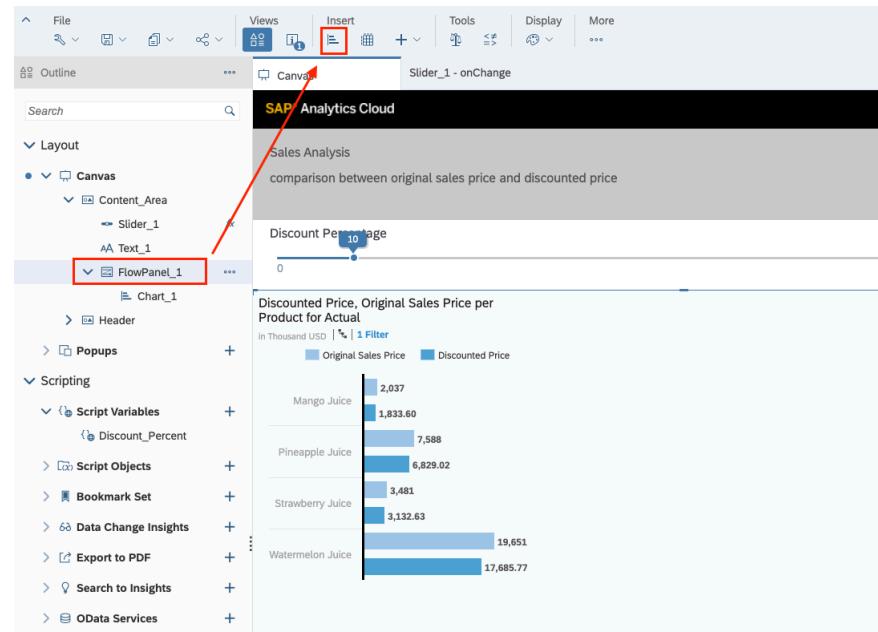


16. As you can see the result, we have now built a chart based in 1 dimension which is Product (only Juice Product) and 2 measures:



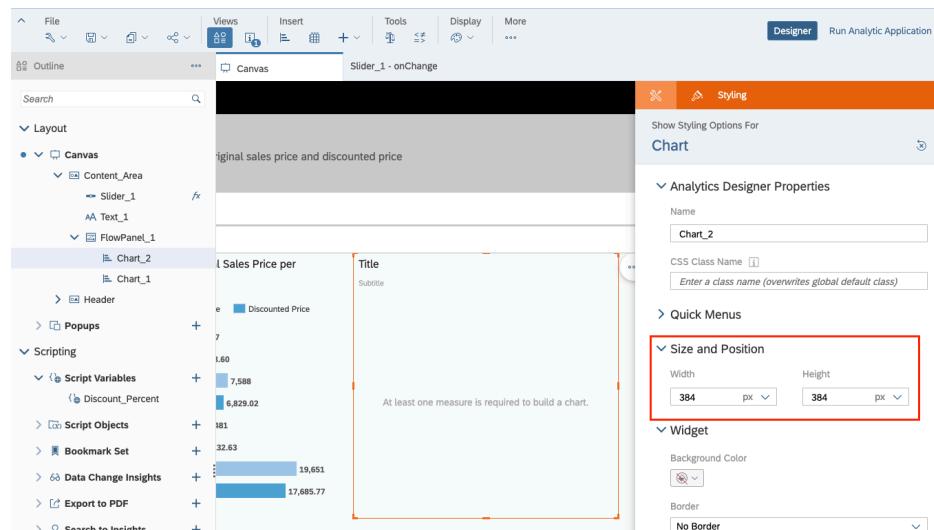
17. Let's add another type of chart a **Tree Map Chart**. Click on the icon **Chart** in order to add another chart widget.

Don't forget to select always the **FlowPanel\_1** from the outline before adding the widget.



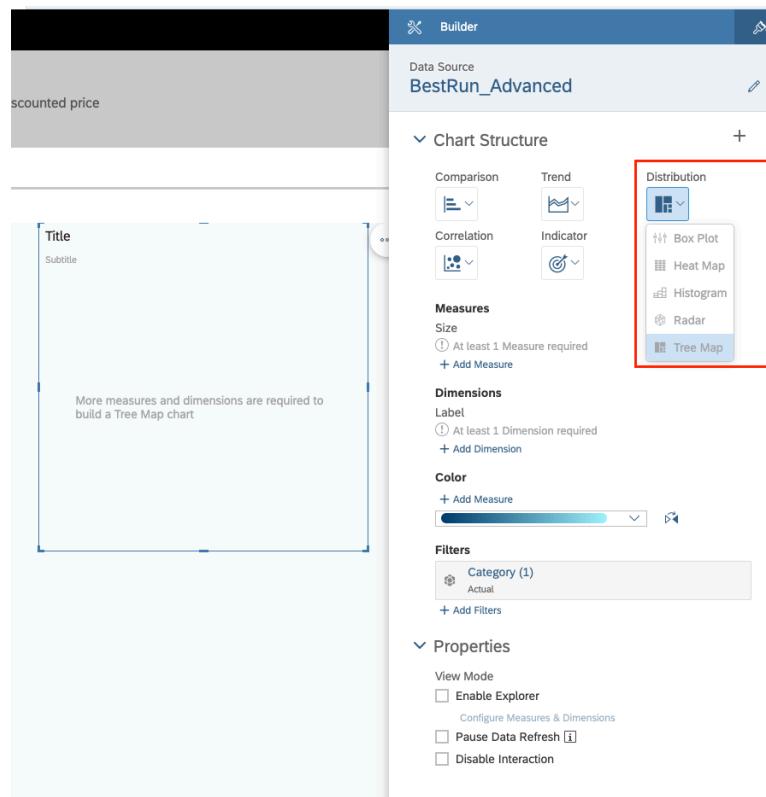
18. Keep the default layout configurations for the chart widget:

**Width:** 384 px  
**Height:** 384 px

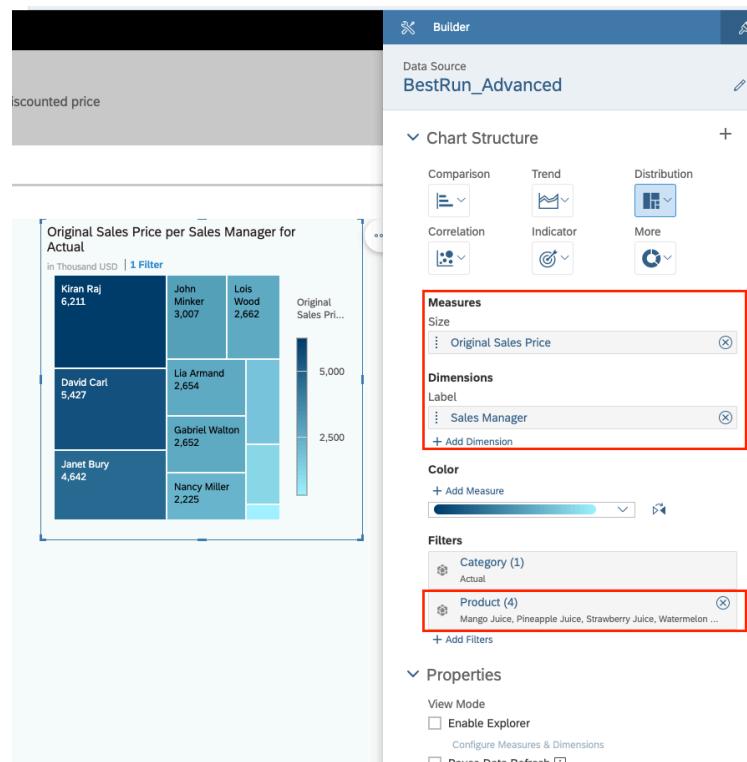


19. Toggle to the **Builder** panel in order to modify the default **Chart Structure** to Tree Map.

**Under Chart Structure**  
**→ Distribution → Tree Map**



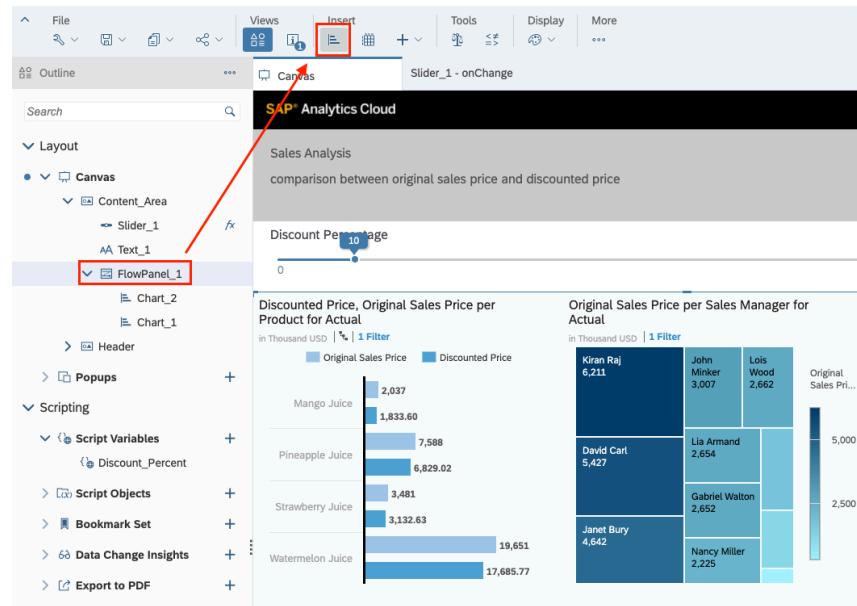
20. Add **Original Sales Price** as a Measure



21. Add **Sales Manager** as a Dimension
22. Add the same filter as the previous chart.  
**(Steps: 15/16/17)**

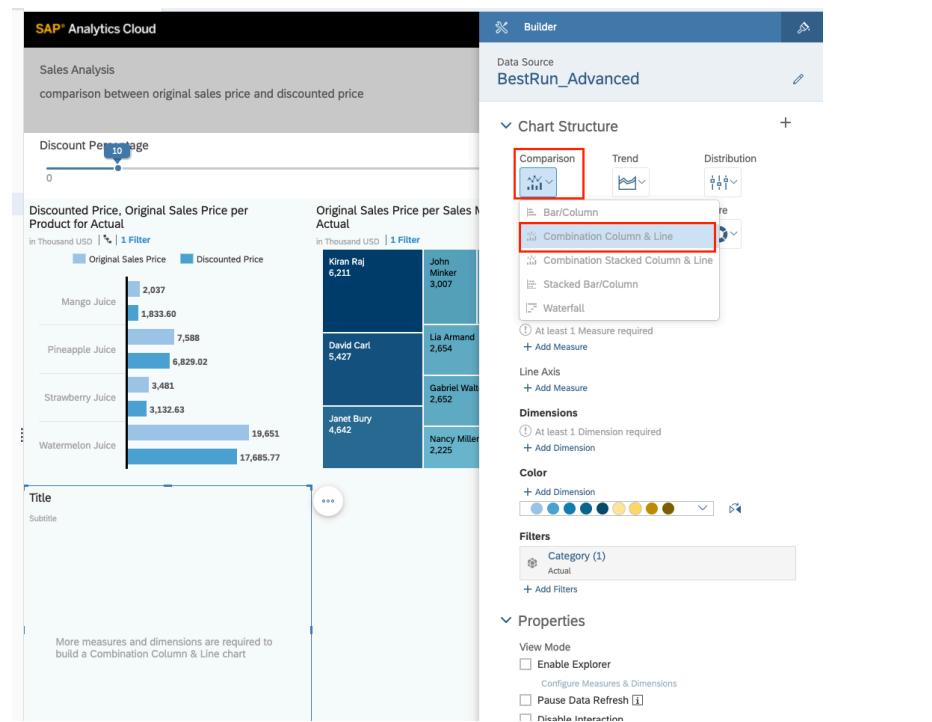
23. Let's add another type of chart a **Combination Column & Line**. Click on the icon **Chart** in order to add another chart widget.

Don't forget to select always the **FlowPanel\_1** from the outline before adding the widget.



24. Toggle to the **Builder** panel in order to modify the default **Chart Structure** to **Combination Column & Line**.

**Under Chart Structure → Comparison → Combination Column & Line**

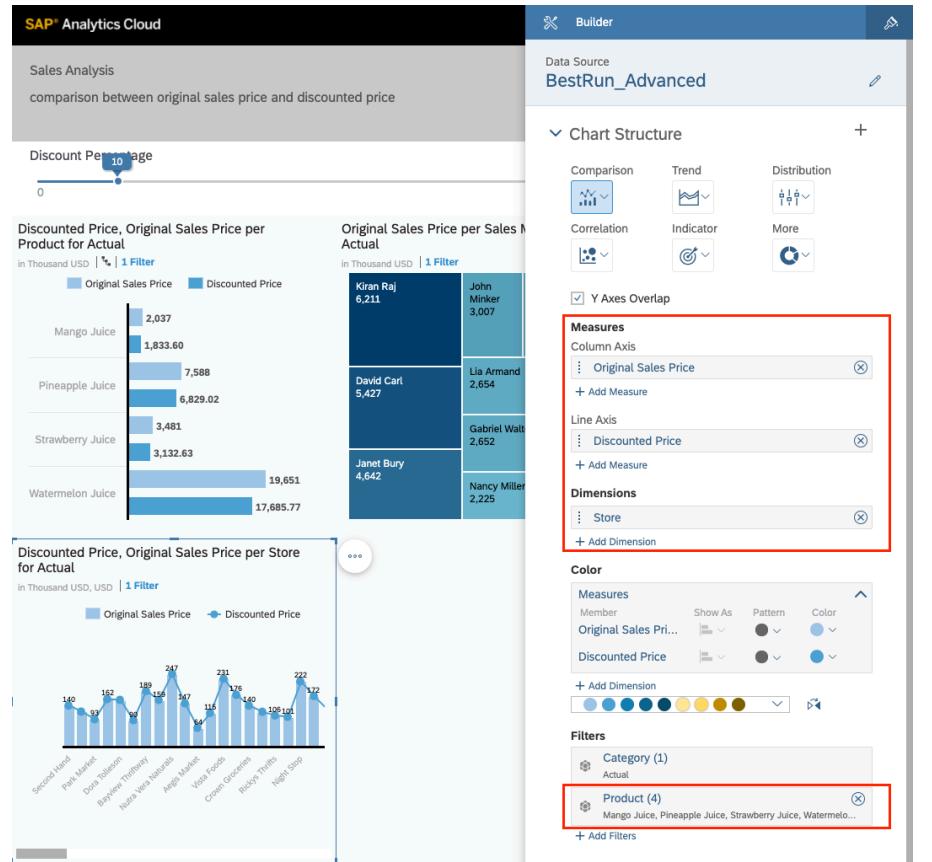


25. Add **Original Sales Price** as a Measure (Column Axis)

26. Add **Discount Price** as a Calculated Measure (Line Axis)

27. Add **Store** as a Dimension

28. Add the same filter as the previous chart.  
**(Steps: 15/16/17)**

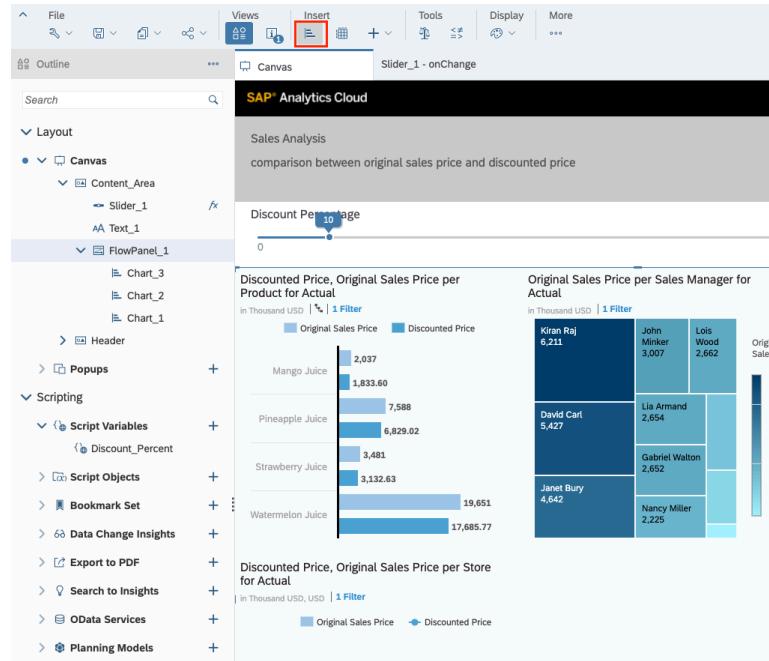


29. Let's add another type of chart a **Stacked Bar/Column**. Click on the icon **Chart** in order to add another chart widget.

Don't forget to select always the **FlowPanel\_1** from the outline before adding the widget.

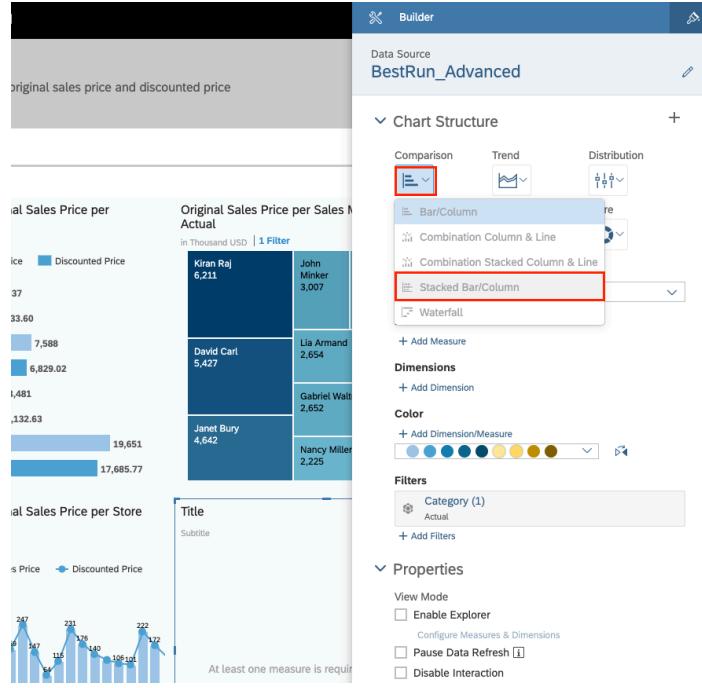
30. Keep the default layout configurations for the chart widget as the previous charts:

**Width:** 384 px  
**Height:** 384 px



31. Toggle to the **Builder** panel in order to modify the default **Chart Structure** to **Stacked Bar/Column**.

32. Under **Chart Structure** → **Comparison** → **Stacked Bar/Column**.



33. Add **Original Sales Price** as the first Measure.

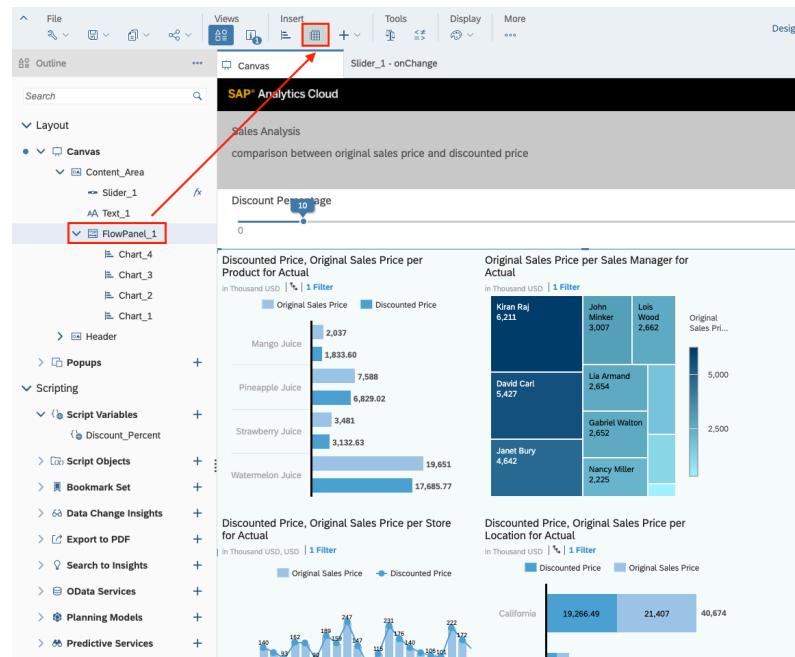
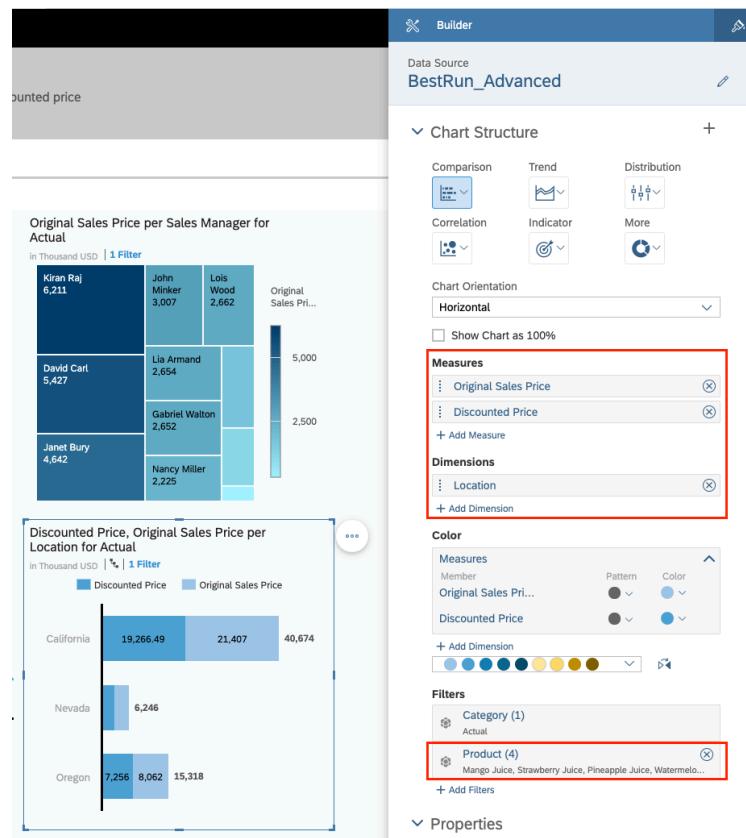
34. Add **Discount Price** as a Calculated Measure.

35. Add **Location** as a Dimension

36. Add the same filter as the previous chart.  
**(Steps: 15/16/17)**

37. Let's add now a Table. Click on the icon **Table** under the insert toolbar.

Don't forget to select always the **FlowPanel\_1** from the outline before adding the widget.



38. Toggle to the **Styling** panel to set some properties for the Table.

39. Define the layout configurations for the **Table** widget:

**Width:** 750 px  
**Height:** 368 px

40. Switch to **Builder Panel**, we are going to add measure(s) and dimension(s) to this table

41. Let's start by adding a dimension to our Table

The screenshot shows the Tableau Builder interface with the following details:

- Data Source:** BestRun\_Advanced
- Table Structure:**
  - Responsive / flexible column width
  - Arrange totals / parent nodes below
  - Beta table (i)
- Rows:** + Add Measures/Dimensions (highlighted with a red box)
- Columns:** Account (1 Model Members | 1 Story Calculations) (with a delete icon)
- Filters:**
  - Account (2) Original Sales Price, Discounted Price
  - Category (1) public.Actual (Actual)
- Properties:**
  - View Mode
  - Enable Explorer

42. Select **Product** as a dimension.

The screenshot shows the Tableau Builder interface with the following details:

- Data Source:** BestRun\_Advanced
- Table Structure:**
  - Responsive / flexible column width
  - Arrange totals / parent nodes below
  - Beta table (i)
- Dimensions:**
  - Search
  - + Create Calculated Dimension
  - Product** (highlighted with a red box)
  - Cross Calculations
  - Account
  - Category
  - Currency
  - Location
  - Sales Manager
  - Store
  - Time
  - Version
- Properties:**
  - Pause Data Refresh (i)
  - Disable Interaction

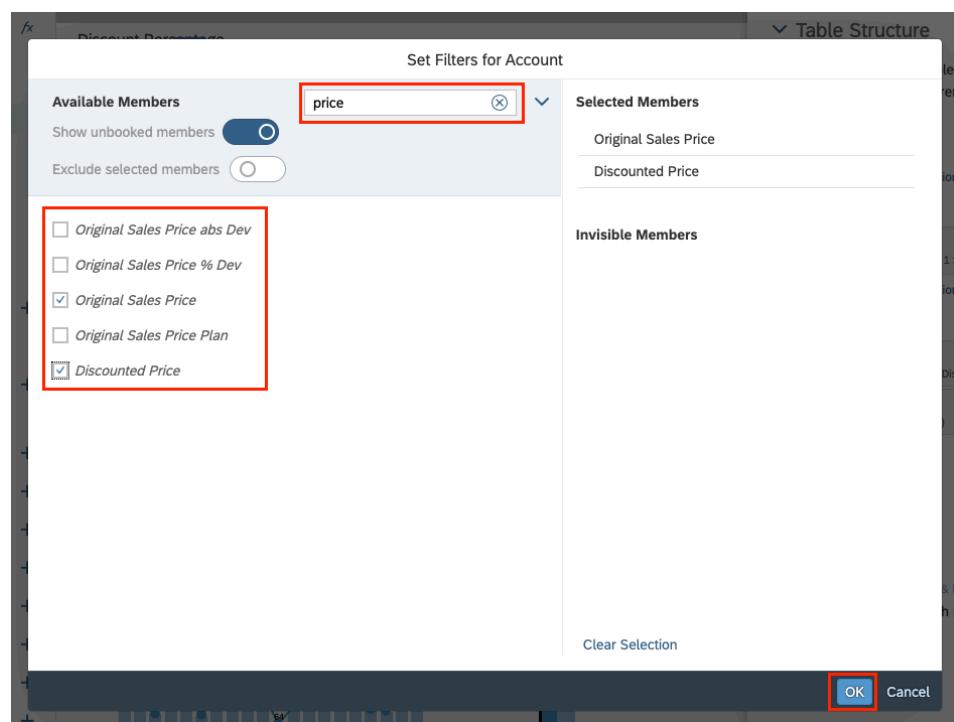
43. Let's manage the **Columns** of the table by clicking the **filter icon** in order to select the proper Members and calculation.

The screenshot shows the Tableau Builder interface with a data source named "BestRun\_Advanced". Under "Table Structure", there are sections for "Rows" and "Columns". The "Columns" section is highlighted with a red box and contains a "Filter" icon (a small square with a diagonal line) next to the "Account" field. Below this, there are buttons for "Add Measures/Dimensions" and "Add Filters". The "Filters" section lists "Account (2)" and "Category (1)".

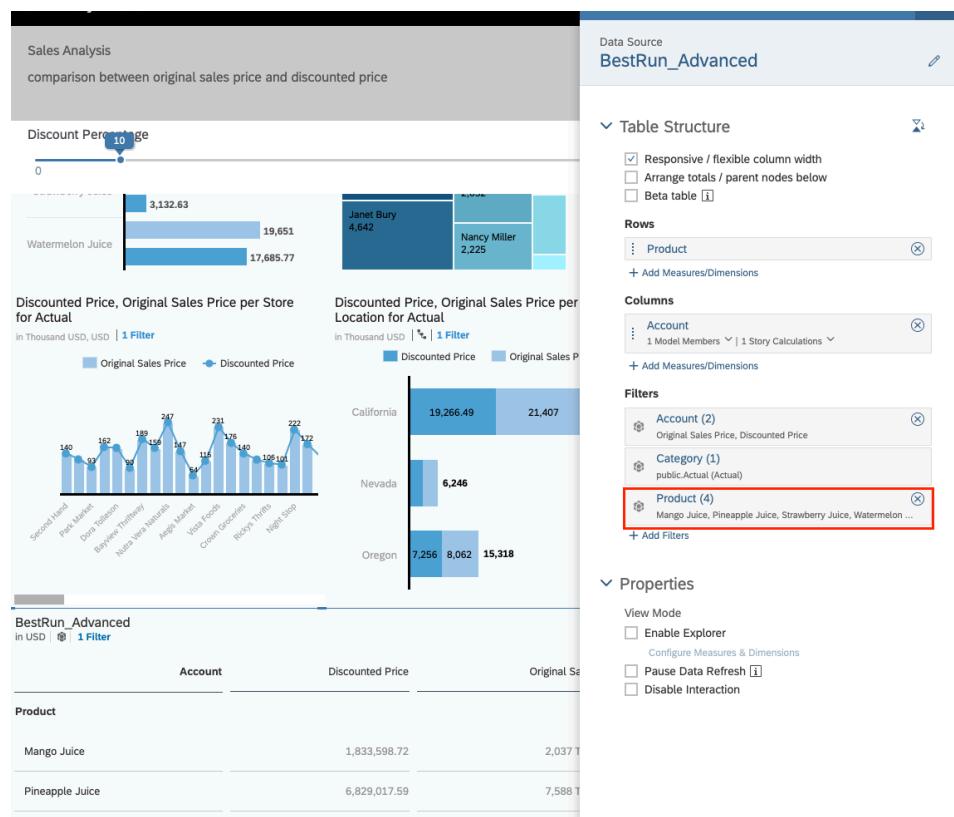
44. First of all, delete the default selected member.

The screenshot shows the "Set Filters for Account" dialog box. It has sections for "Available Members", "Selected Members", and "Invisible Members". In the "Selected Members" section, there is a list containing "Discount abs Dev" and a delete icon (a small square with a diagonal line). The "Available Members" section contains a list of various members and calculations, with "Discount abs Dev" checked. The "Invisible Members" section is empty. At the bottom, there are "OK" and "Cancel" buttons.

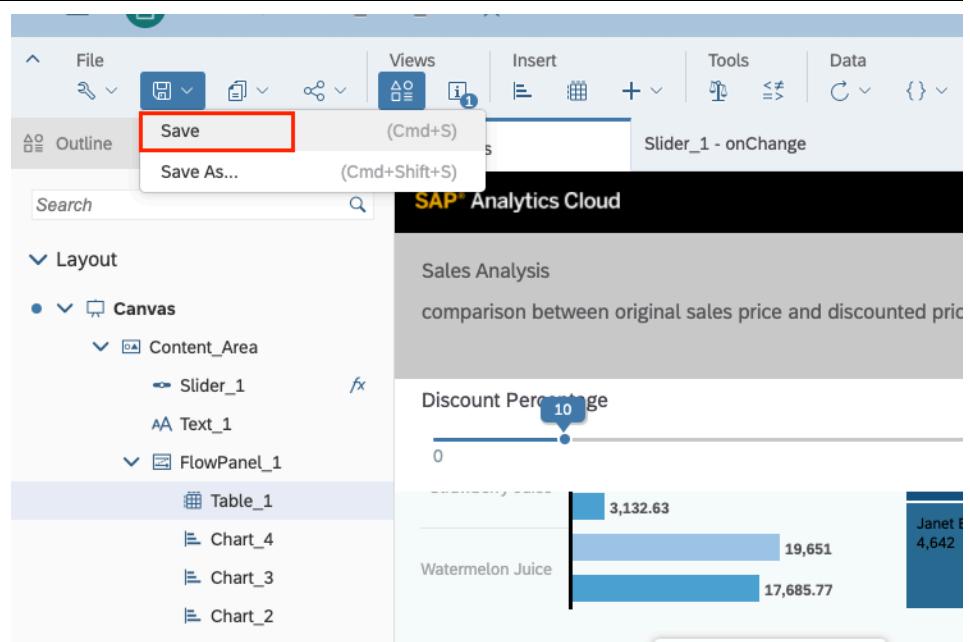
45. Use the **search bar** to search for the **Original Sales Price** and **Discounted Price**, then click **OK** to confirm.



46. Add the same filter as the previous Charts to the Table.  
**Check steps 15/16/17**

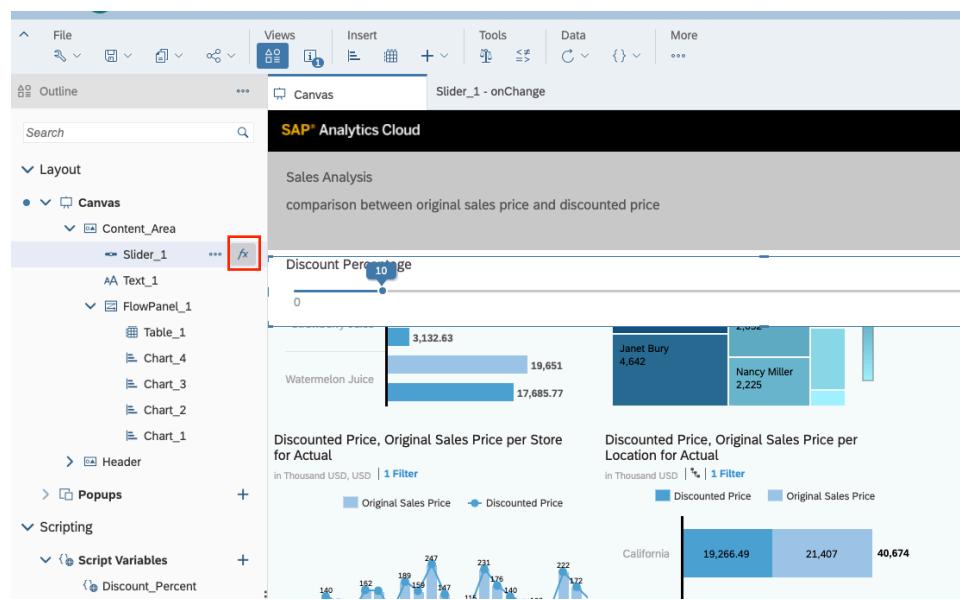


47. Click on the **Floppy Disc** icon and choose **Save**.

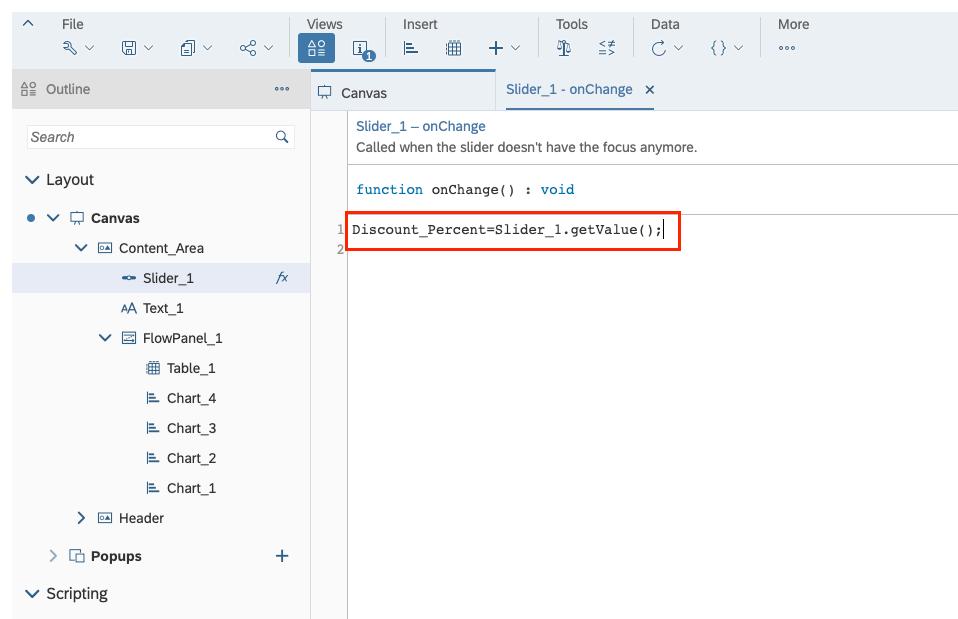


48. As next, let's define the **onChange** event for the Slider widget that we defined in the beginning of the exercise.

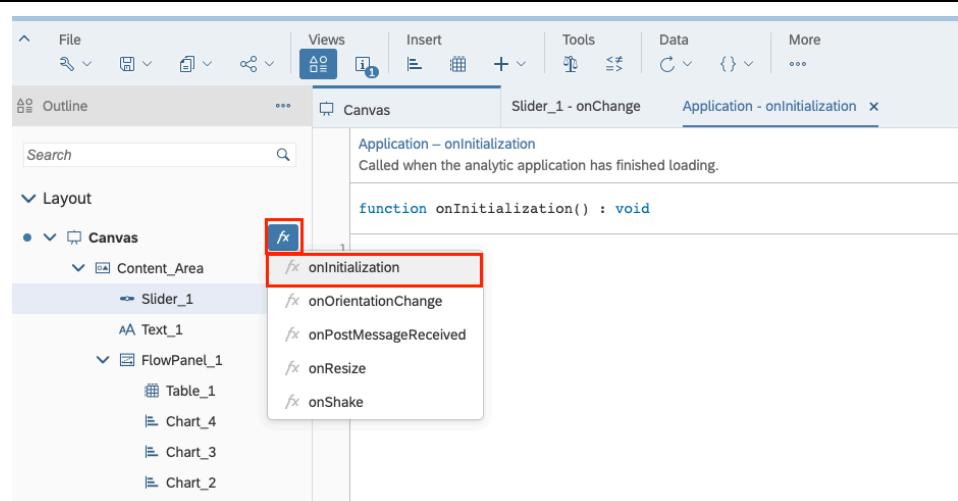
Please position on the **Slider\_1** and click on the **fx** icon.



49. You can see that a Tab **Slider\_1.onChange** is open next to the Canvas Tab. Within this Tab we will add the Script. The coding should cover the following logic:  
Assign the value of the current selection of the slider to script variable **Discount\_Percent**. You can use the scripting wizard to write the command or you can copy and paste this line of code:
- ```
Discount_Percent=Slider_1.getValue();
```

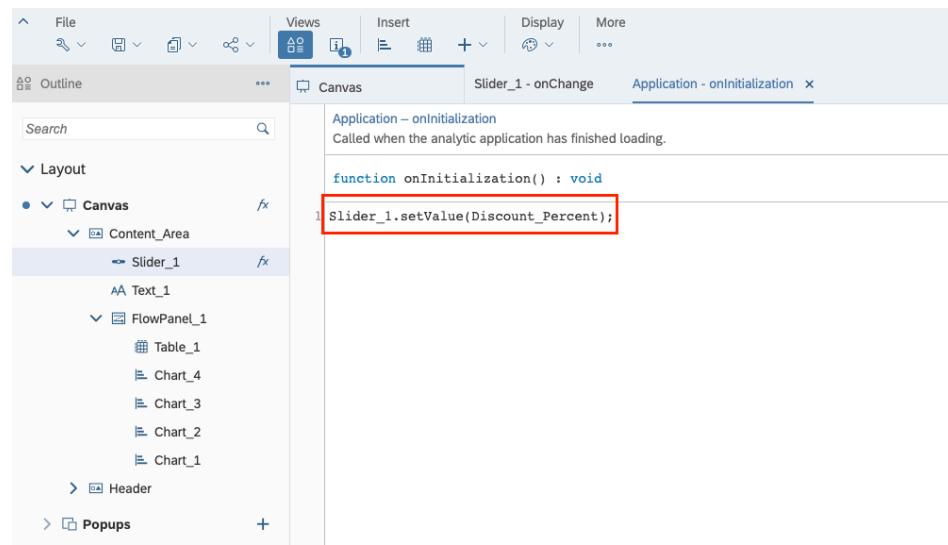


50. The last step is tracking what happens when the application is initialized. This is done through the **onInitialization** function of the Canvas itself. To write the responsible script, please hover over the **Canvas** in the layout outline panel and click on the **fx** icon, then select **onInitialization**.

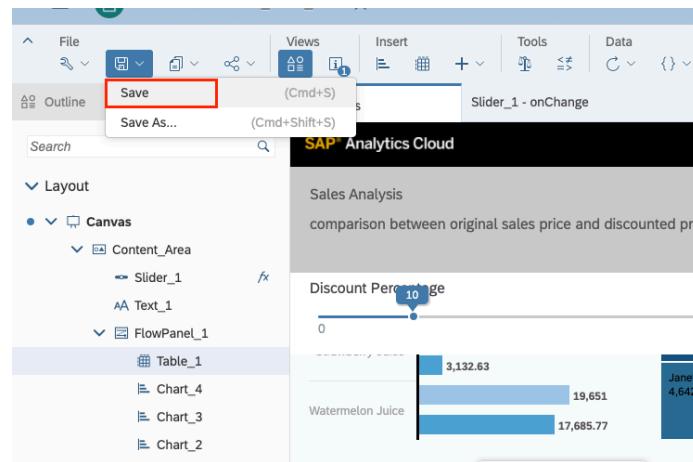


51. Now we need to define the script to set the variable value in the slider.

Enter the code below:  
**Slider\_1.setValue(Discount\_Percent);**

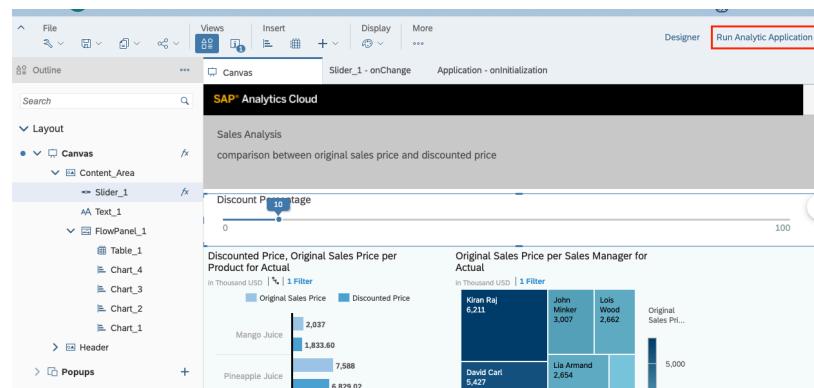


52. Click on the Floppy Disc icon and choose Save.



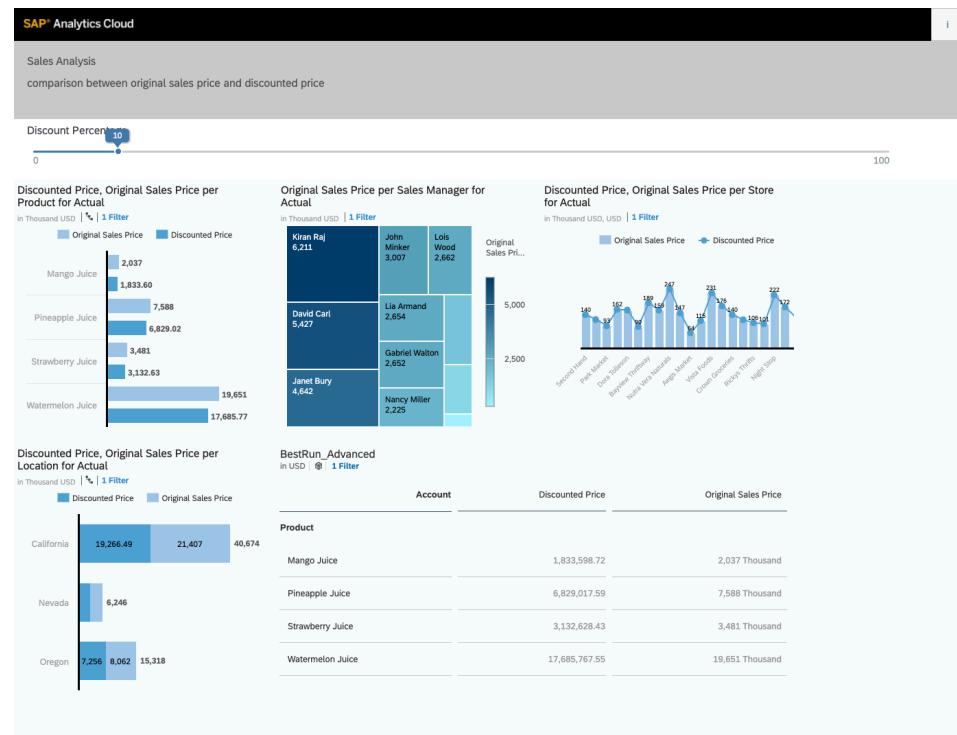
53. It's time to show how this application is working at runtime.

For that please click Run Analytical Application.

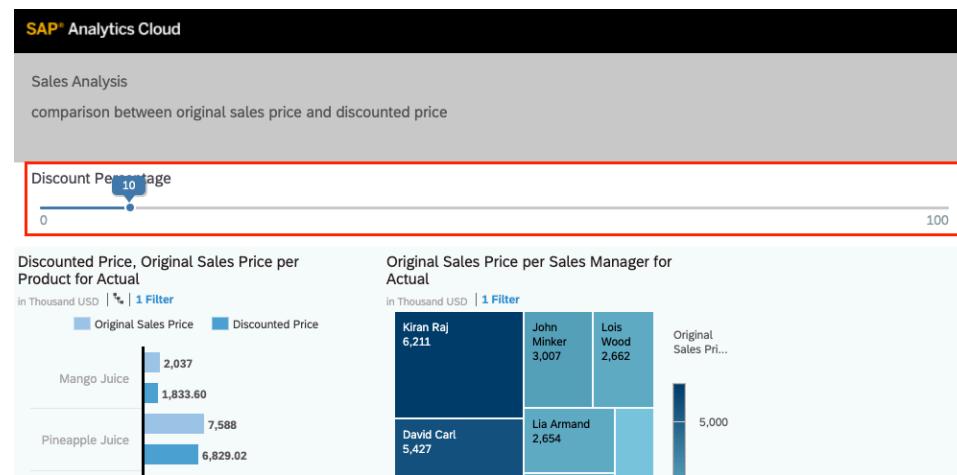


54. As you can see, you built an Analytical application which composed of different widgets (different type of Charts/ Table/ Slider) within a responsive frame which is the Flow Panel feature.

Try to reduce the size of your browser in order to see how the Flow Panel that we used will behave.



55. As next, please change the value in the Slider and check the charts and table behaviors.

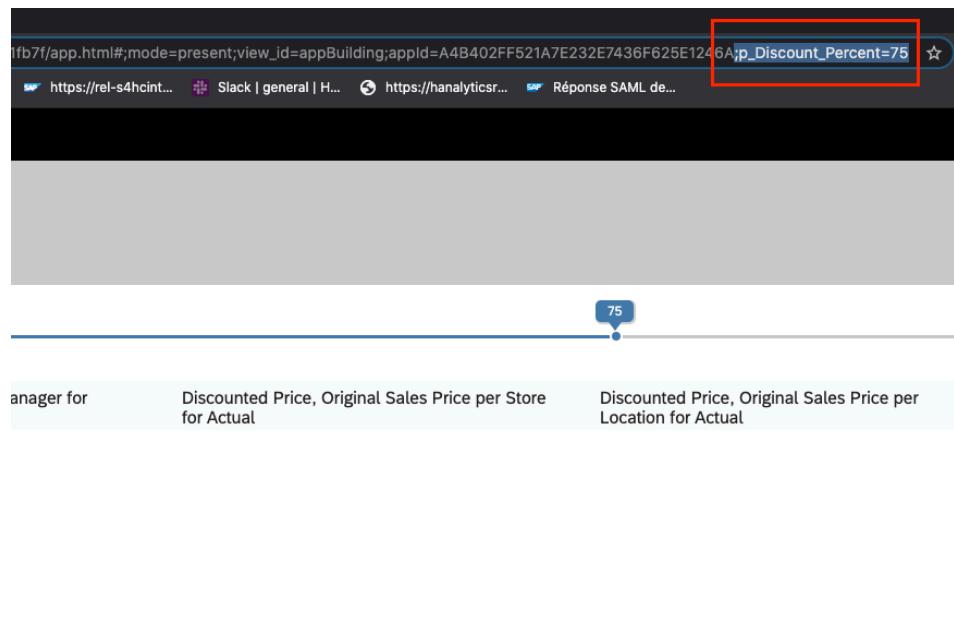


56. You can try to enter at the end of the URL:

`:p_Discount_Percent=75`

In the definition of the script variable, we selected the option **Expose Variable via URL Parameter**. This make possible to set the variable value via URL. In the URL of the analytic application, just enter a new parameter that starts with “`p_`” and is followed by the script variable's name:

`:p_Discount_Percent=75`



### Congratulations! You just finished the part 2 of Exercise 1.

Theme provides application designer an efficient and reusable way to define the style of the application. When you choose a theme, you change the app's look and feel instantly.  
In the **Part 3** of the exercise, we will show you how to use a theme.

## Part 3: Changing Theme

1. Go back to the application and click on the **Themes** icon

SAP Analytics Cloud

Sales Analysis  
comparison between original sales price and discounted price

Discount Percentage: 10

Original Sales Actual  
in Thousand USD | 1 Filter

Mango Juice: Original Sales Price 2,037, Discounted Price 1,833.60

Kiran Raj 6,211

2. Choose **Browse for More Themes ...**

SAP Analytics Cloud

Sales Analysis  
comparison between original s

Discount Percentage: 10

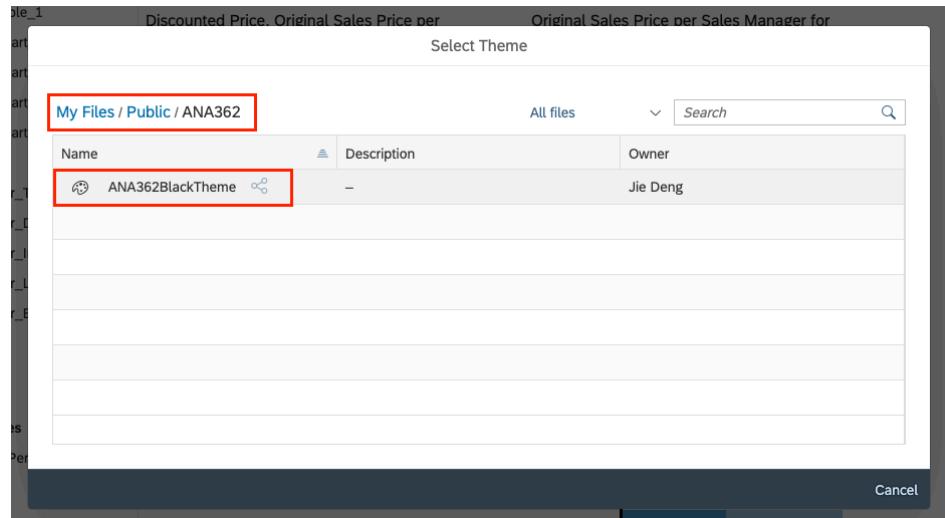
Original Sales Actual  
in Thousand USD | 1 Filter

Mango Juice: Original Sales Price 2,037, Discounted Price 1,833.60

Kiran Raj 6,211

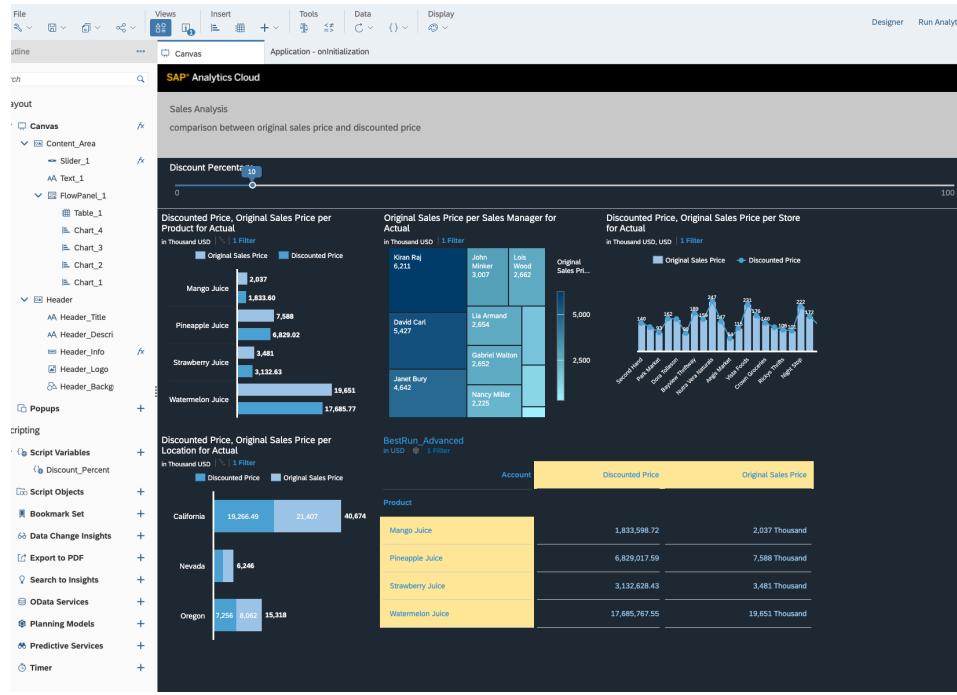
**Browse for More Themes...**

3. Redirect to the folder **My Files → Public → ANA362**, then choose the theme **ANA362BlackTheme**

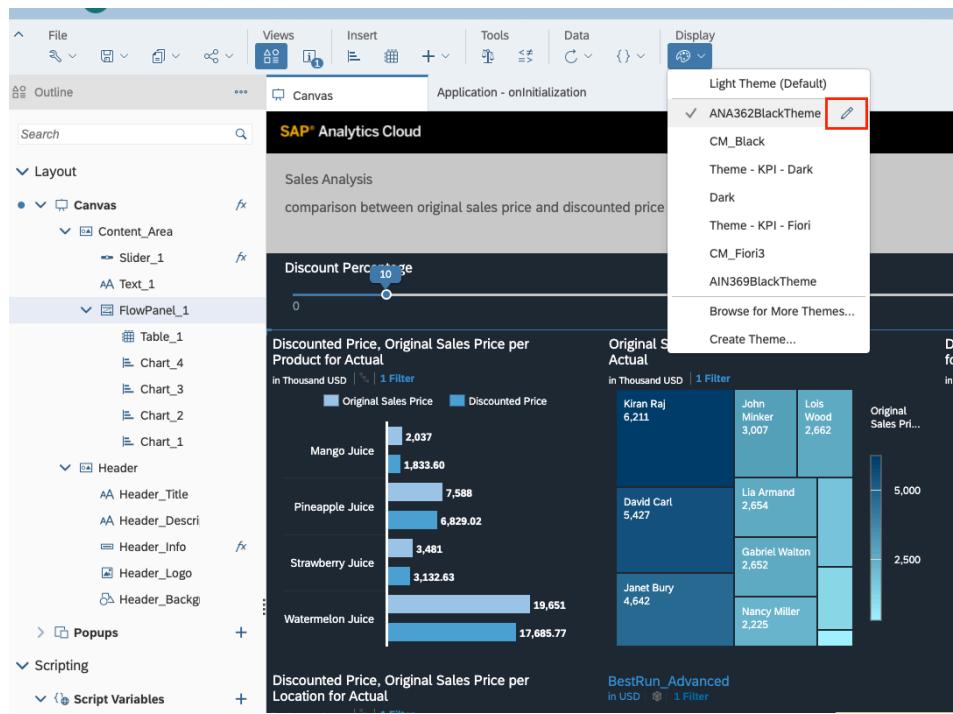


4. You can see now that the Theme changed.

As next we will show which themes preferences can be changed

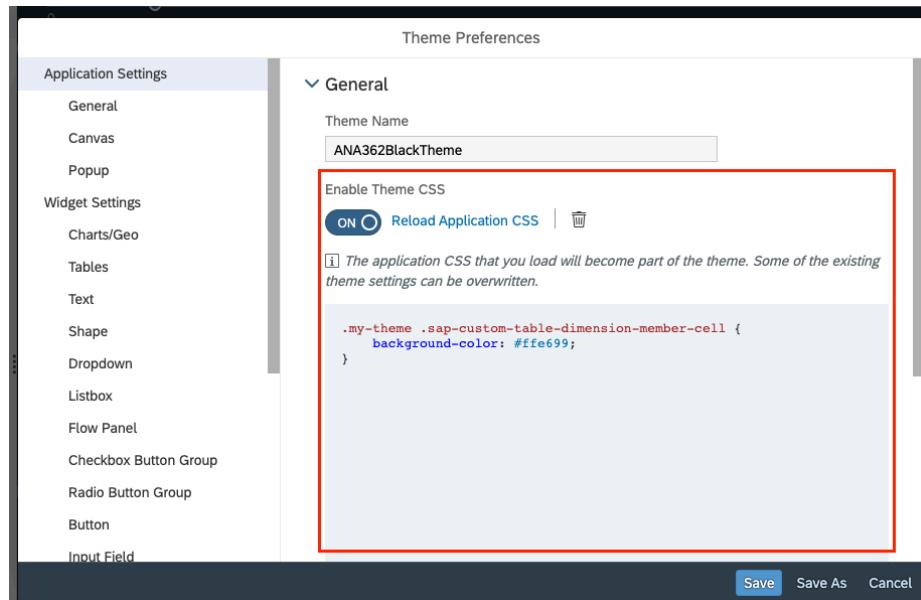


5. Hover over the active theme **ANA362BlackTheme** and click on the **Pencil** icon

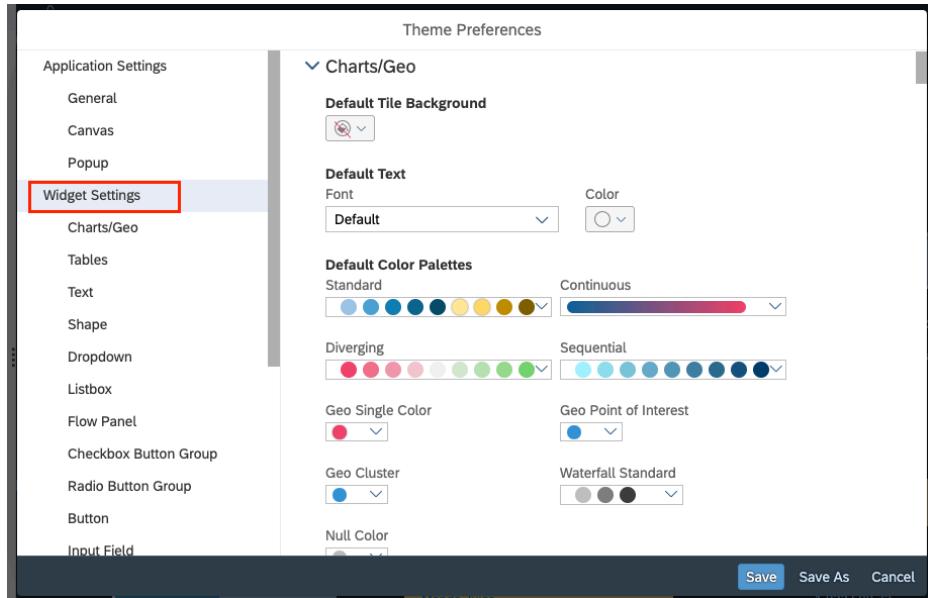


6. As you can see, the **CSS script** used for this theme. With **Analytical Application** you have the possibility to write and load your specific **CSS script**.

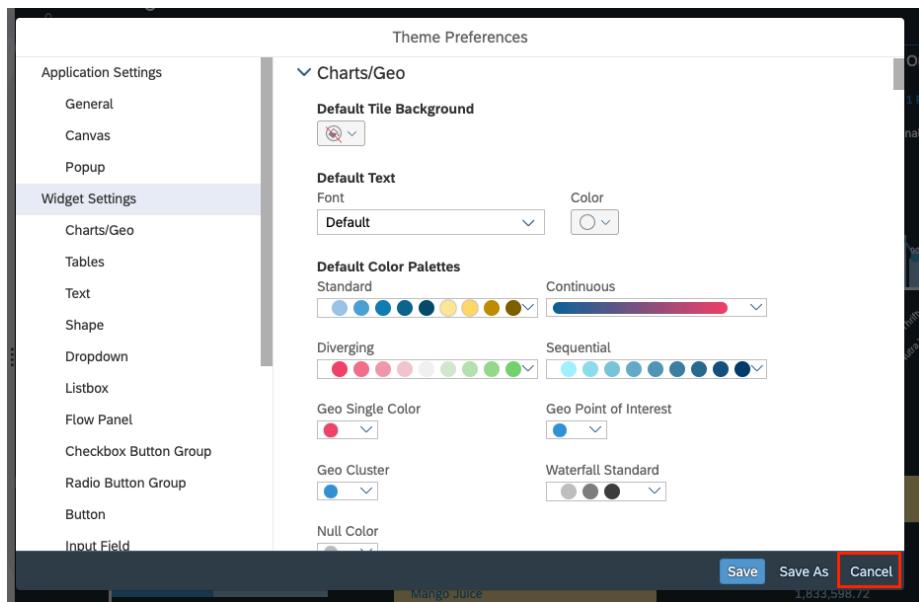
You can as well explore the different options available; you can change the **Theme Preferences** of any **widgets** available on your application.



7. For example, choose **Widget Settings** in order to see the different **Theme Preferences**.

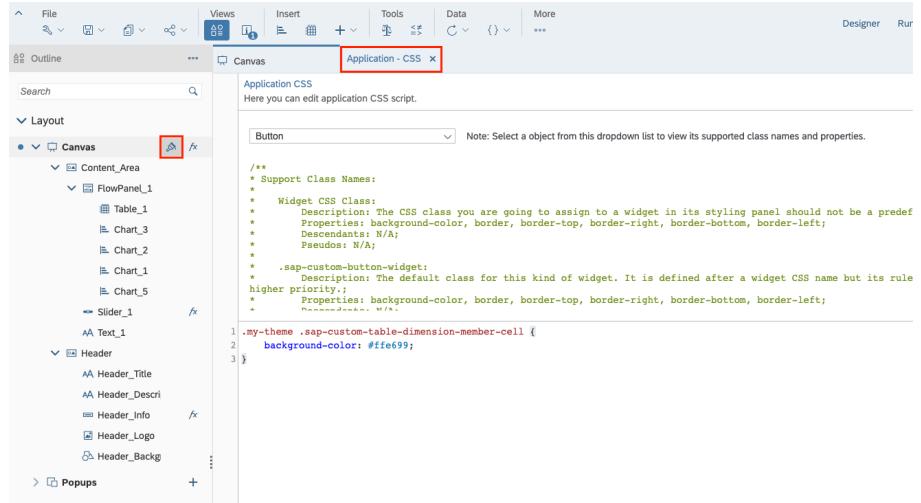


8. Click **Cancel** to exit the Theme Preferences pop up

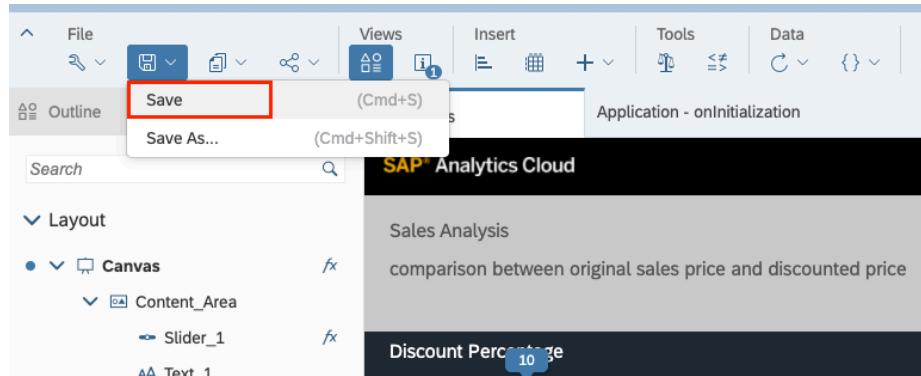


9. By hover over the **Canvas** from the **outline** panel you can click the **styling icon** in order to edit the **application CSS** code.

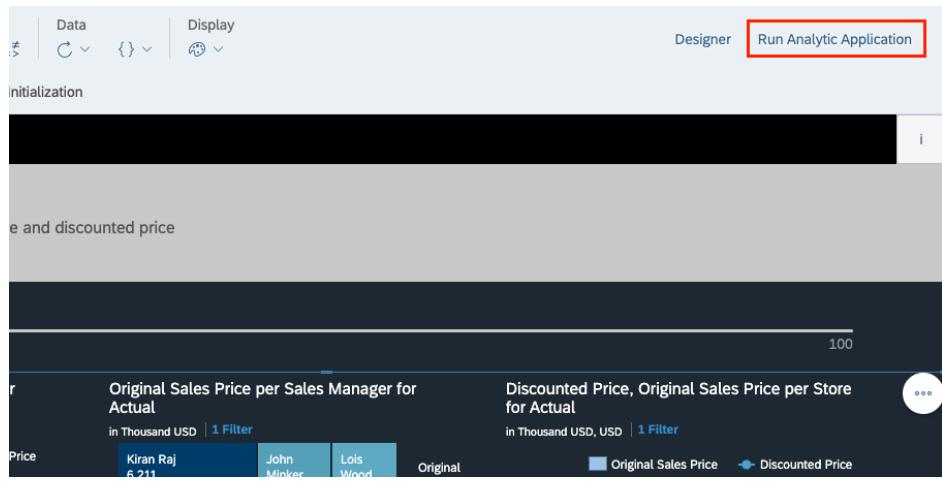
As shown a new tab called **Application-CSS** appeared where you can start editing the **CSS code** based on the supported Class Name.



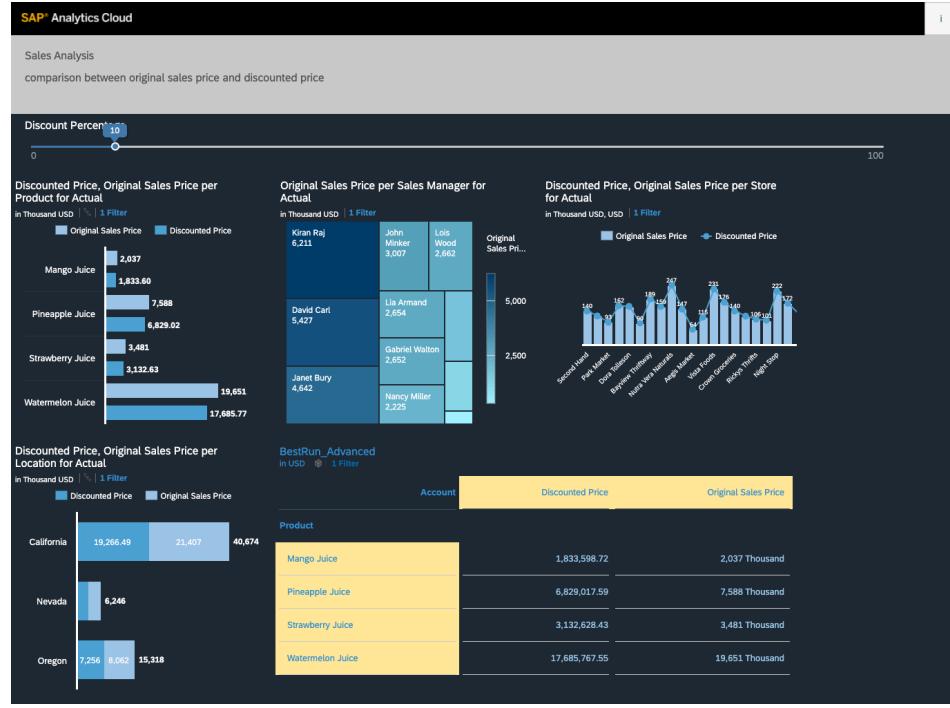
10. Click on the **Floppy Disc** icon and select **Save**.



11. Click **Run Analytic Application** to see how the new selected theme look like in the runtime.



12. Congratulations! You added the theme successfully.



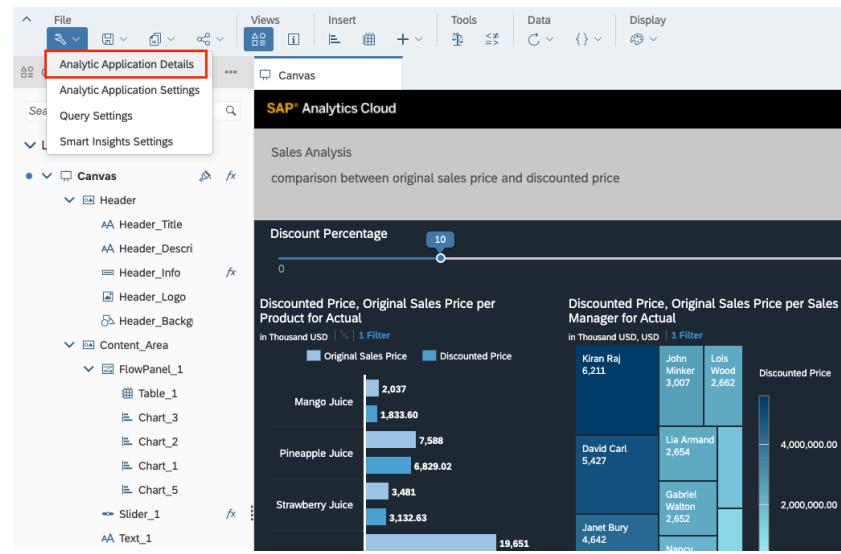
**Congratulations! You just finished the part 3 of Exercise 1.**

In the **Part 4** of the exercise you will download the SAC mobile app from Apple Store and run it with your Device.

## Part 4: Download SAC app from Apple Store and run the application

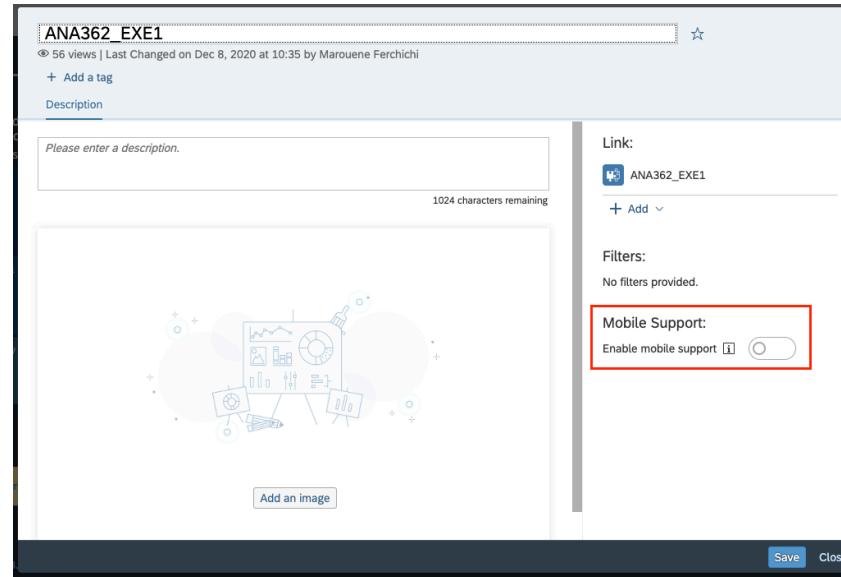
- Before downloading the SAC app, let's set the ANA362\_EXE1 as mobile enabled.

- Under the setting icon, click **Analytical Application Details**

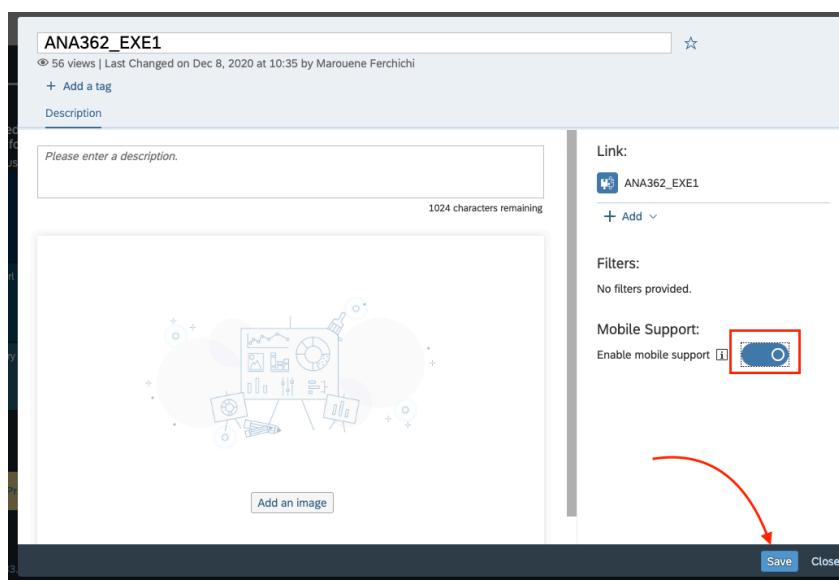


- As shown the Mobile support option is **disabled**.

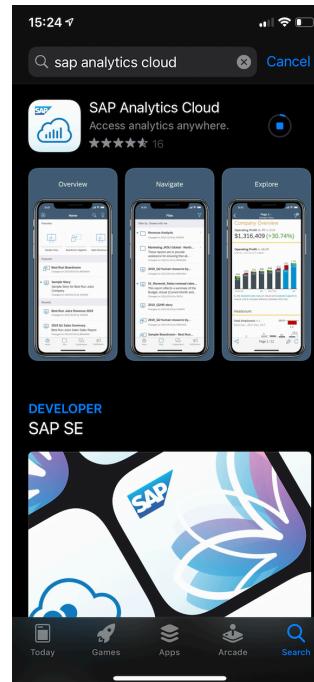
- Please **enable** the mobile support



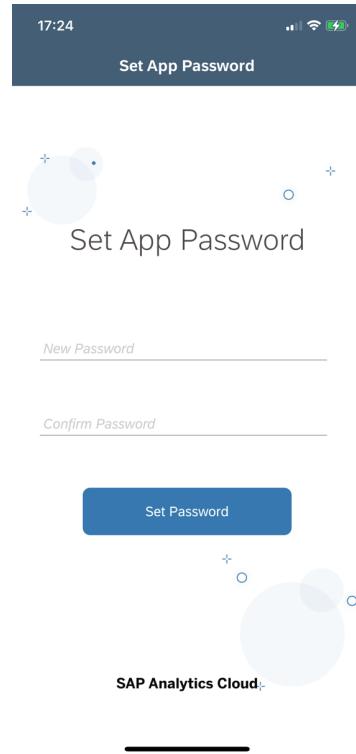
5. After the **enable** the Mobile support, click **Save**.



6. Now let's test the application with your mobile device.  
7. Open the Apple Store and search for the SAC app and press Download, the same as you will download another app with your phone.



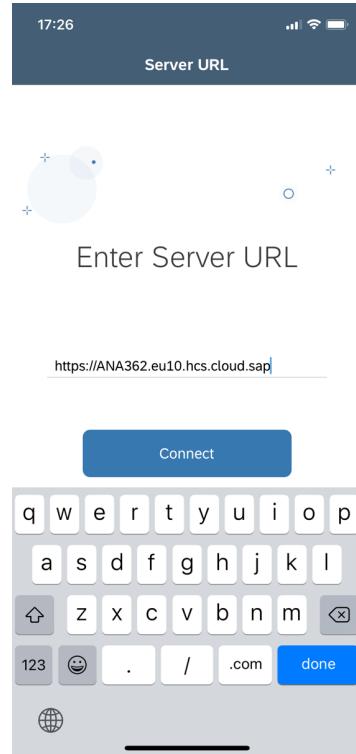
- 
8. As next, open the app and create a **Password** for the app



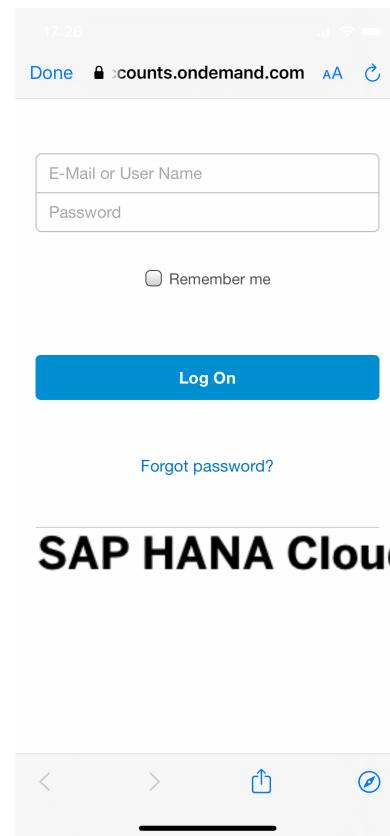
- 
9. Next step you have to enter the **Server URL**:

<https://ANA362.eu10.hcs.cloud.sap>

Press **Connect** afterwards.



- 
10. Use your Credential to connect to the tenant in order to find your application and run it on your Mobile.



- 
11. Redirect to your files and select the application in order to run it in the mobile device.



**Congratulations! You just finished the part 4 of Exercise 1.**

---

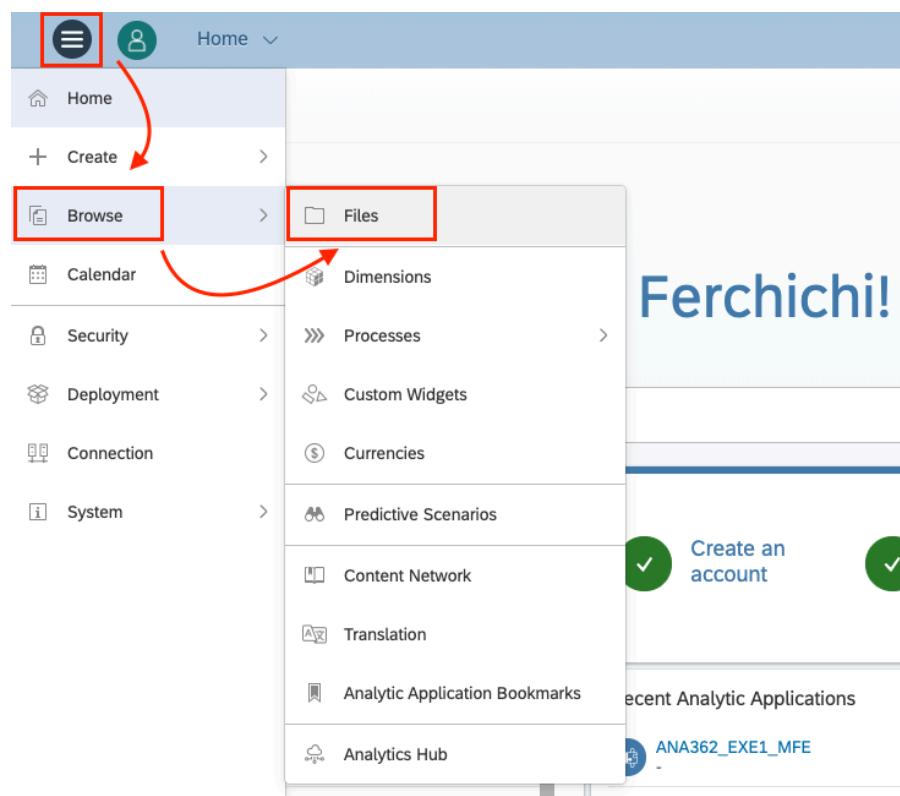
In the **Part 5** of the exercise you will use the Post Message capability of the SAC to embed your analytical application into a HTML Page.

---

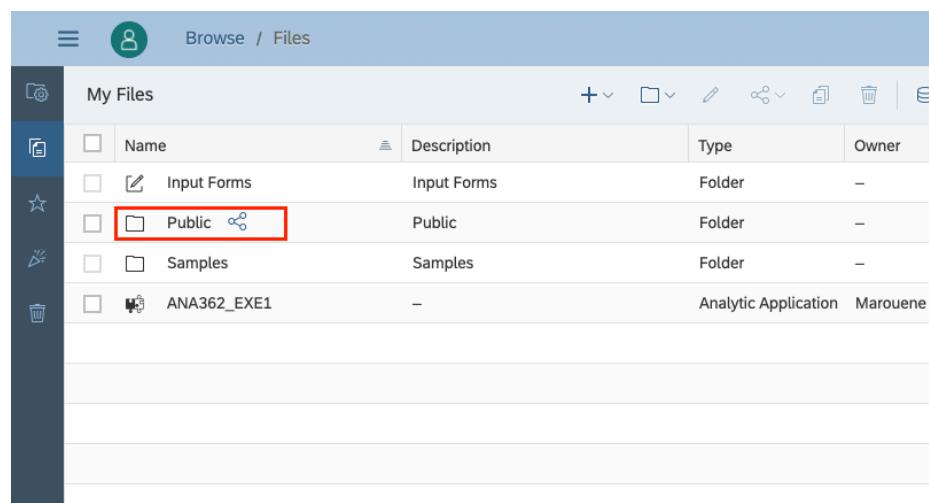
## Part 5: Embedding into a HTML page

- Let's open the ANA362 to download our TXT file that we will modify and use for this part of the exercise.

Click **File**

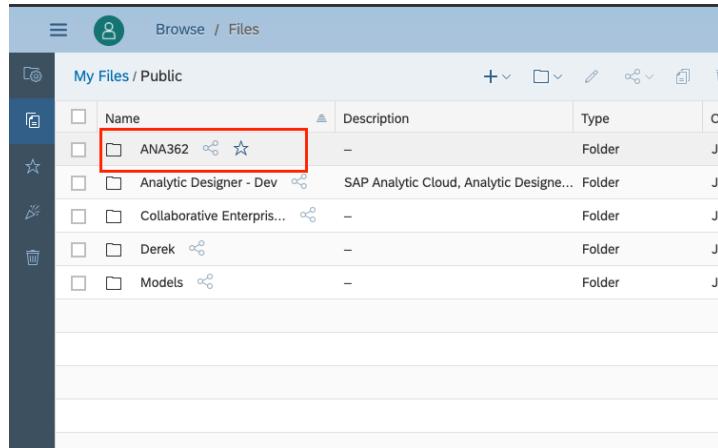


- Then open the Folder **Public**

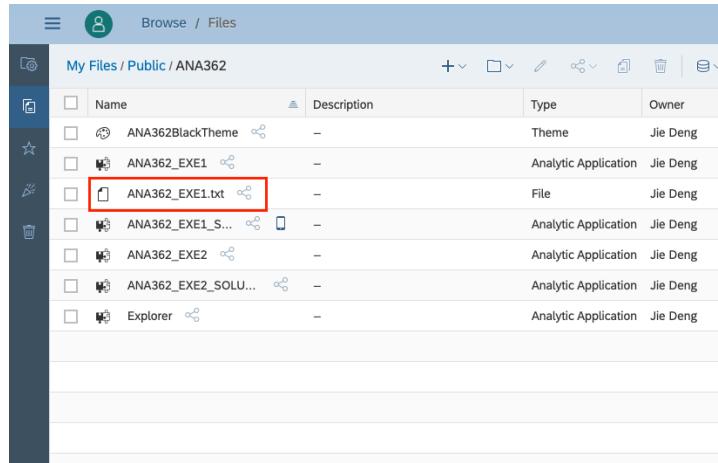


## ANA362

3. Click the folder **ANA362** to open it.

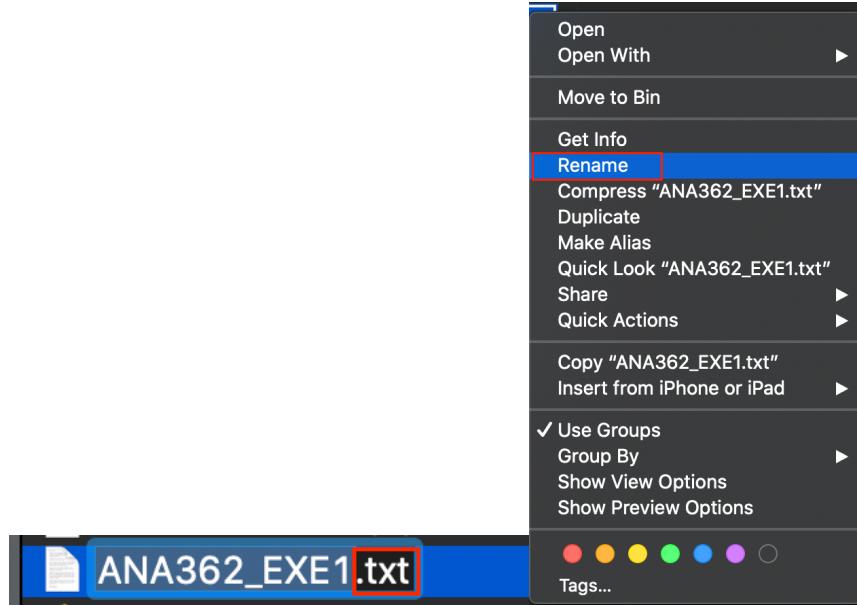


4. Just Click the **ANA362\_EXE1.txt** to download the Text file.

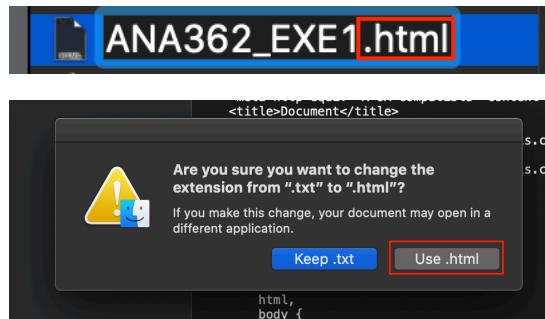


5. As next, when the download is finish, open the folder where the file was downloaded and change the extension from: **.txt** to **.html**

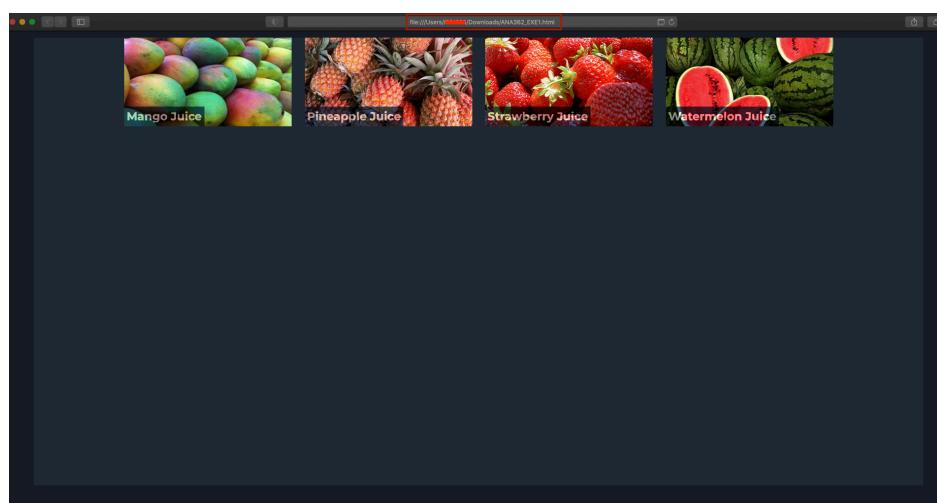
6. Right Click on the file and select **Rename**



7. Delete .txt and replace it by .html



8. Afterwards Double Click on the **HTML** file in order to open it with your browser.



9. Here is a HTML page. We should now integrate the analytic application created before into this HTML page.

**There are bi-directional integrations:**

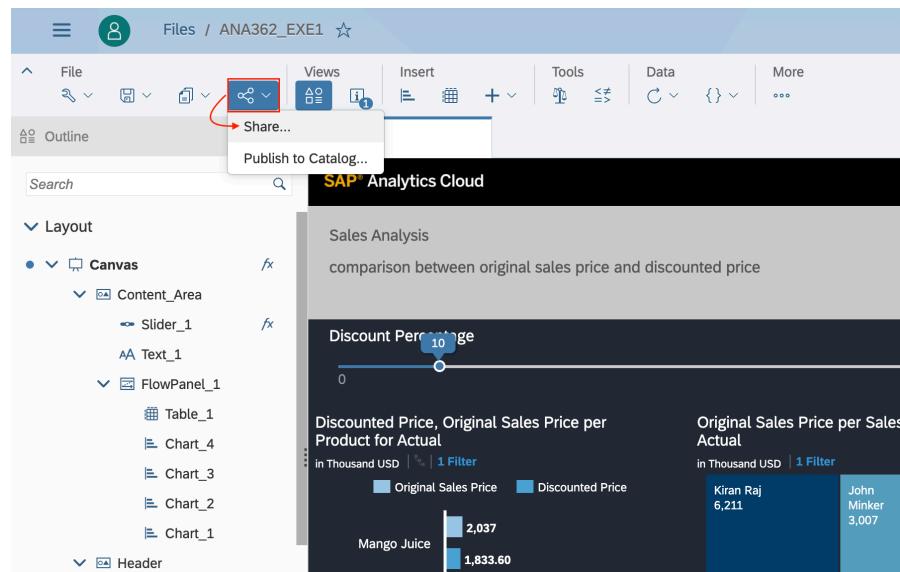
- 1) If user selects the juice type on the html page, this information should be used for filtering the charts in analytical applications
- 2) If user changes the discount percentage value by slider in analytical application, then this parameter should be transferred to html page for calculating the discounted price

**For these interactions, you need to do the following steps:**

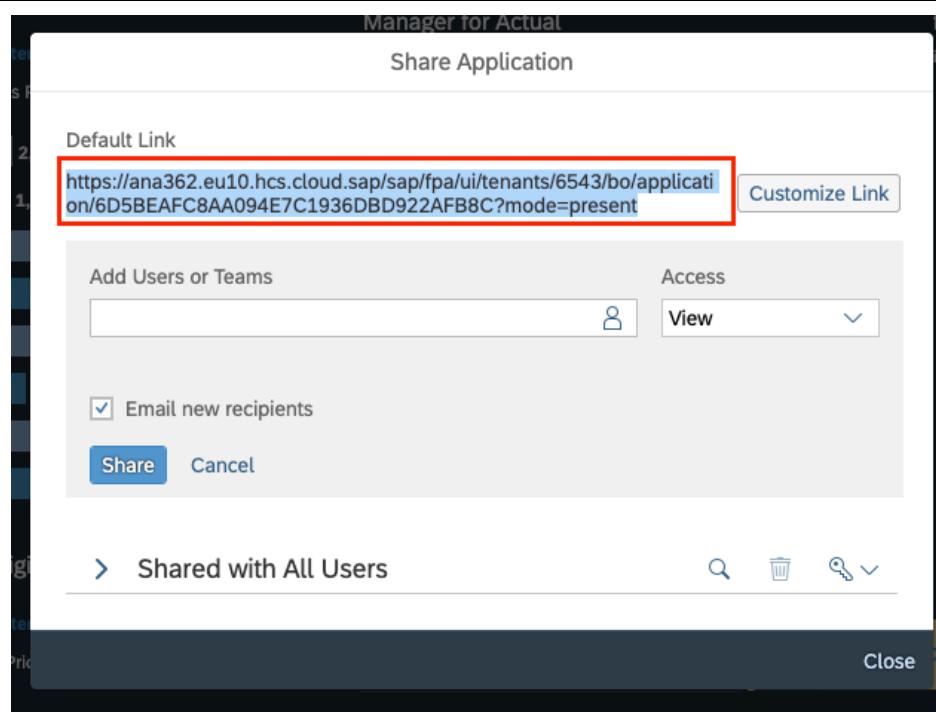
- 1) adapt the application name in html page
- 2) add the PostMessageReceived event in the analytical application
- 3) add the post message API to On Click event for the Slider (pass the parameter discount percentage to html page)
- 4) add the post message API to OnInitialization event (pass the original sales price to html page)

10. Go back to your SAP Analytics Cloud application.

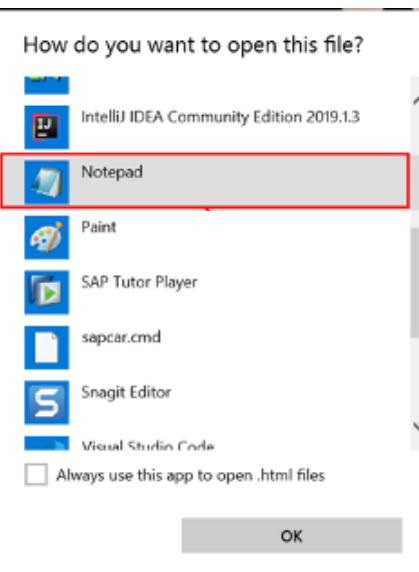
Click **Share** to get the application URL



11. Copy the application ID of your application  
**ANA362\_EXE1**



12. As next, go back to the **HTML** file and right click on: **ANA362.html**



13. Click on “**Open With**” and select **Notepad** or another file editor that you have.

14. Please scroll down until you find the default **URL**.

15. Delete the existing one and replace it with your **URL** that you copied in **STEP 10**, press **CTRL + V** and **CTRL + S** to save the changes.

16. As next, copy the tenant path:  
<https://ana362.eu10.hcs.cloud.sap> from the Application ID that you used in **Step 14**

```

<!-->
<figure id="watermelon">
  
  <figcaption>
    <h3>Watermelon Juice</h3>
  </figcaption>
  <span></span>
  <span></span>
</figure>
</div>
<iframe style="width: 1440px; height: 1120px;" src="https://ana362.eu10.hcs.cloud.sap/sap/fpa/ui/tenants/6543/bo/application/A4BD4AFF4A4D60DB6F7C3DE7AD3C69A3?mode=present"></iframe>
<script>
  document.addEventListener("readystatechange", (ev) => {
    if (document.readyState === "interactive") {
      addClickHandler();
    }
  });

```

```

<span></span>
<span></span>
</div>
<iframe style="width: 1440px; height: 1120px;" src="https://ana362.eu10.hcs.cloud.sap/sap/fpa/ui/tenants/6543/bo/application/6D5BEAFC8AA094E7C1936DBD922AFB8C?mode=present"></iframe>
<script>
  document.addEventListener("readystatechange", (ev) => {
    if (document.readyState === "interactive") {
      addClickHandler();
    }
  });

```

17. Paste the tenant path in the place of the 'XXXXXXXXXXXX'

```
</div>
<iframe style="width: 1440px; height: 1120px;"  
src="https://ana362.eu10.hcs.cloud.sap/sap/fpa/ui/tenants/6543/bo/application/6D5BEAFC8AA094E7C1936DBD922AFB8C?mode=p  
</div>

<script>  
document.addEventListener("readystatechange", (ev) => {  
    if (document.readyState === "interactive") {  
        addClickHandler();  
    }  
});  
  
function addClickHandler() {  
    document.querySelector("#mango").addEventListener("click", () => {  
        window.frames[0].postMessage('mango', 'XXXXXXXXXXXXXX');  
    });  
    document.querySelector("#pineapple").addEventListener("click", () => {  
        window.frames[0].postMessage('pineapple', 'XXXXXXXXXXXXXX');  
    });  
    document.querySelector("#strawberry").addEventListener("click", () => {  
        window.frames[0].postMessage('strawberry', 'XXXXXXXXXXXXXX');  
    });  
    document.querySelector("#watermelon").addEventListener("click", () => {  
        window.frames[0].postMessage('watermelon', 'XXXXXXXXXXXXXX');  
    });  
  
    window.addEventListener("message", e => {  
        // ...  
    });  
}
```

18. Your file should look like that after your paste the tenant path

```
if (document.readyState === "interactive") {  
    addClickHandler();  
}  
});  
  
function addClickHandler() {  
    document.querySelector("#mango").addEventListener("click", () => {  
        window.frames[0].postMessage('mango', 'https://ana362.eu10.hcs.cloud.sap');  
    });  
    document.querySelector("#pineapple").addEventListener("click", () => {  
        window.frames[0].postMessage('pineapple', 'https://ana362.eu10.hcs.cloud.sap');  
    });  
    document.querySelector("#strawberry").addEventListener("click", () => {  
        window.frames[0].postMessage('strawberry', 'https://ana362.eu10.hcs.cloud.sap');  
    });  
    document.querySelector("#watermelon").addEventListener("click", () => {  
        window.frames[0].postMessage('watermelon', 'https://ana362.eu10.hcs.cloud.sap');  
    });  
  
    window.addEventListener("message", e => {  
        if (e.data[0] !== "{}") return;  
    });  
}
```

19. Save the changes by pressing **CTRL + S**

20. Go back to your Analytical application  
**ANA362\_EXE1**

21. Open the **fx** of the **Canvas** in order to write some script

22. Select **OnPostMessageReceive** d event.

The screenshot shows the SAP Analytics Cloud interface with the 'Sales Analysis' dashboard. In the canvas editor, a context menu is open over the 'onPostMessageReceived' event node. The menu items include: comparison between original sales price and discounted price, onInitialization, onOrientationChange, **onPostMessageReceived**, onResize, and onShake. The 'onPostMessageReceived' item is highlighted with a red box and has a red arrow pointing to it from the left.

23. In this part, we are passing the message (mango, pineapple etc.) into the application to apply the product name as filter to chart.

Please enter (or copy & paste) the following scripting for the **onPostMessageReceivedEvent**:

```

if (message === "mango") {
    Chart_1.getDataSource().setDimensionFilter("Product_3e315003an", "[Product_3e315003an].[ABC].&[PD14]");
    Chart_2.getDataSource().setDimensionFilter("Product_3e315003an", "[Product_3e315003an].[ABC].&[PD14]");
    Chart_3.getDataSource().setDimensionFilter("Product_3e315003an", "[Product_3e315003an].[ABC].&[PD14]");
    Chart_4.getDataSource().setDimensionFilter("Product_3e315003an", "[Product_3e315003an].[ABC].&[PD14]");
    Table_1.getDataSource().setDimensionFilter("Product_3e315003an", "[Product_3e315003an].[ABC].&[PD14]");
}

if (message === "pineapple") {
    Chart_1.getDataSource().setDimensionFilter("Product_3e315003an", "[Product_3e315003an].[ABC].&[PD15]");
    Chart_2.getDataSource().setDimensionFilter("Product_3e315003an", "[Product_3e315003an].[ABC].&[PD15]");
    Chart_3.getDataSource().setDimensionFilter("Product_3e315003an", "[Product_3e315003an].[ABC].&[PD15]");
    Chart_4.getDataSource().setDimensionFilter("Product_3e315003an", "[Product_3e315003an].[ABC].&[PD15]");
    Table_1.getDataSource().setDimensionFilter("Product_3e315003an", "[Product_3e315003an].[ABC].&[PD15]");
}

if (message === "strawberry") {
    Chart_1.getDataSource().setDimensionFilter("Product_3e315003an", "[Product_3e315003an].[ABC].&[PD18]");
    Chart_2.getDataSource().setDimensionFilter("Product_3e315003an", "[Product_3e315003an].[ABC].&[PD18]");
    Chart_3.getDataSource().setDimensionFilter("Product_3e315003an", "[Product_3e315003an].[ABC].&[PD18]");
    Chart_4.getDataSource().setDimensionFilter("Product_3e315003an", "[Product_3e315003an].[ABC].&[PD18]");
    Table_1.getDataSource().setDimensionFilter("Product_3e315003an", "[Product_3e315003an].[ABC].&[PD18]");
}

if (message === "watermelon") {
    Chart_1.getDataSource().setDimensionFilter("Product_3e315003an", "[Product_3e315003an].[ABC].&[PD16]");
    Chart_2.getDataSource().setDimensionFilter("Product_3e315003an", "[Product_3e315003an].[ABC].&[PD16]");
    Chart_3.getDataSource().setDimensionFilter("Product_3e315003an", "[Product_3e315003an].[ABC].&[PD16]");
    Chart_4.getDataSource().setDimensionFilter("Product_3e315003an", "[Product_3e315003an].[ABC].&[PD16]");
    Table_1.getDataSource().setDimensionFilter("Product_3e315003an", "[Product_3e315003an].[ABC].&[PD16]");
}

```

The screenshot shows the 'Application - onPostMessageReceived' script editor. The code block contains logic to filter data sources based on the message received ('mango', 'pineapple', 'strawberry', or 'watermelon'). The code uses the 'setDimensionFilter' method on various chart and table data sources to apply filters like '[Product\_3e315003an].[ABC].&[PD14]' through '[ABC].&[PD16]'. A red box highlights the entire code block.

```

function onPostMessageReceived(message: string, origin: string) : void
1 if (message === "mango") {
2
3     Chart_1.getDataSource().setDimensionFilter("Product_3e315003an", "[Product_3e315003an].[ABC].&[PD14]");
4     Chart_2.getDataSource().setDimensionFilter("Product_3e315003an", "[Product_3e315003an].[ABC].&[PD14]");
5     Chart_3.getDataSource().setDimensionFilter("Product_3e315003an", "[Product_3e315003an].[ABC].&[PD14]");
6     Chart_4.getDataSource().setDimensionFilter("Product_3e315003an", "[Product_3e315003an].[ABC].&[PD14]");
7     Table_1.getDataSource().setDimensionFilter("Product_3e315003an", "[Product_3e315003an].[ABC].&[PD14]");
8 }
9
10 if (message === "pineapple") {
11     Chart_1.getDataSource().setDimensionFilter("Product_3e315003an", "[Product_3e315003an].[ABC].&[PD15]");
12     Chart_2.getDataSource().setDimensionFilter("Product_3e315003an", "[Product_3e315003an].[ABC].&[PD15]");
13     Chart_3.getDataSource().setDimensionFilter("Product_3e315003an", "[Product_3e315003an].[ABC].&[PD15]");
14     Chart_4.getDataSource().setDimensionFilter("Product_3e315003an", "[Product_3e315003an].[ABC].&[PD15]");
15     Table_1.getDataSource().setDimensionFilter("Product_3e315003an", "[Product_3e315003an].[ABC].&[PD15]");
16 }
17
18 if (message === "strawberry") {
19     Chart_1.getDataSource().setDimensionFilter("Product_3e315003an", "[Product_3e315003an].[ABC].&[PD18]");
20     Chart_2.getDataSource().setDimensionFilter("Product_3e315003an", "[Product_3e315003an].[ABC].&[PD18]");
21     Chart_3.getDataSource().setDimensionFilter("Product_3e315003an", "[Product_3e315003an].[ABC].&[PD18]");
22     Chart_4.getDataSource().setDimensionFilter("Product_3e315003an", "[Product_3e315003an].[ABC].&[PD18]");
23     Table_1.getDataSource().setDimensionFilter("Product_3e315003an", "[Product_3e315003an].[ABC].&[PD18]");
24 }
25
26 if (message === "watermelon") {
27     Chart_1.getDataSource().setDimensionFilter("Product_3e315003an", "[Product_3e315003an].[ABC].&[PD16]");
28     Chart_2.getDataSource().setDimensionFilter("Product_3e315003an", "[Product_3e315003an].[ABC].&[PD16]");
29     Chart_3.getDataSource().setDimensionFilter("Product_3e315003an", "[Product_3e315003an].[ABC].&[PD16]");
30     Chart_4.getDataSource().setDimensionFilter("Product_3e315003an", "[Product_3e315003an].[ABC].&[PD16]");
31     Table_1.getDataSource().setDimensionFilter("Product_3e315003an", "[Product_3e315003an].[ABC].&[PD16]");
32 }

```

#### 24. Let's also add some script on the Slider.

Click **fx**

The screenshot shows the 'Application - onPostMessageReceived' script editor. The code block is identical to the one in the previous screenshot. The 'Slider\_1' node is selected in the layout tree, indicated by a red box around its 'fx' icon. This indicates that the script will be applied to the slider component.

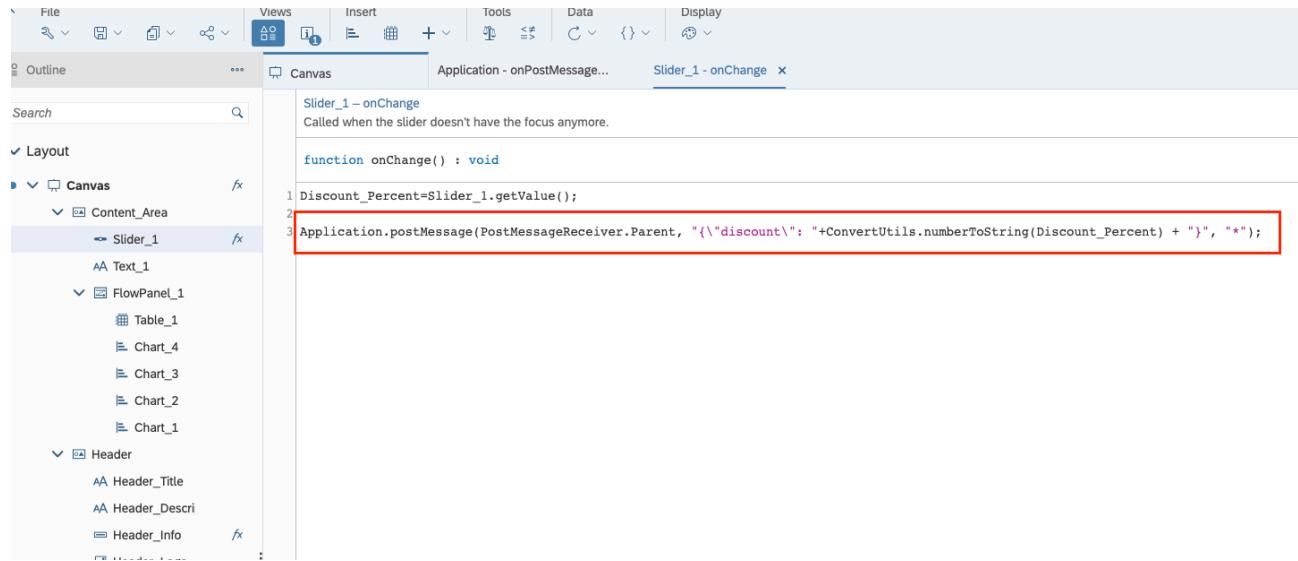
```

function onPostMessageReceived(message: string, origin: string) : void
1 if (message === "mango") {
2
3     Chart_1.getDataSource().setDimensionFilter("Product_3e315003an", "[Product_3e315003an].[ABC].&[PD14]");
4     Chart_2.getDataSource().setDimensionFilter("Product_3e315003an", "[Product_3e315003an].[ABC].&[PD14]");
5     Chart_3.getDataSource().setDimensionFilter("Product_3e315003an", "[Product_3e315003an].[ABC].&[PD14]");
6     Chart_4.getDataSource().setDimensionFilter("Product_3e315003an", "[Product_3e315003an].[ABC].&[PD14]");
7     Table_1.getDataSource().setDimensionFilter("Product_3e315003an", "[Product_3e315003an].[ABC].&[PD14]");
8 }
9
10 if (message === "pineapple") {
11     Chart_1.getDataSource().setDimensionFilter("Product_3e315003an", "[Product_3e315003an].[ABC].&[PD15]");
12     Chart_2.getDataSource().setDimensionFilter("Product_3e315003an", "[Product_3e315003an].[ABC].&[PD15]");
13     Chart_3.getDataSource().setDimensionFilter("Product_3e315003an", "[Product_3e315003an].[ABC].&[PD15]");
14     Chart_4.getDataSource().setDimensionFilter("Product_3e315003an", "[Product_3e315003an].[ABC].&[PD15]");
15     Table_1.getDataSource().setDimensionFilter("Product_3e315003an", "[Product_3e315003an].[ABC].&[PD15]");
16 }
17
18 if (message === "strawberry") {
19     Chart_1.getDataSource().setDimensionFilter("Product_3e315003an", "[Product_3e315003an].[ABC].&[PD18]");
20     Chart_2.getDataSource().setDimensionFilter("Product_3e315003an", "[Product_3e315003an].[ABC].&[PD18]");
21     Chart_3.getDataSource().setDimensionFilter("Product_3e315003an", "[Product_3e315003an].[ABC].&[PD18]");
22     Chart_4.getDataSource().setDimensionFilter("Product_3e315003an", "[Product_3e315003an].[ABC].&[PD18]");
23     Table_1.getDataSource().setDimensionFilter("Product_3e315003an", "[Product_3e315003an].[ABC].&[PD18]");
24 }
25
26 if (message === "watermelon") {
27     Chart_1.getDataSource().setDimensionFilter("Product_3e315003an", "[Product_3e315003an].[ABC].&[PD16]");
28     Chart_2.getDataSource().setDimensionFilter("Product_3e315003an", "[Product_3e315003an].[ABC].&[PD16]");
29     Chart_3.getDataSource().setDimensionFilter("Product_3e315003an", "[Product_3e315003an].[ABC].&[PD16]");
30     Chart_4.getDataSource().setDimensionFilter("Product_3e315003an", "[Product_3e315003an].[ABC].&[PD16]");
31     Table_1.getDataSource().setDimensionFilter("Product_3e315003an", "[Product_3e315003an].[ABC].&[PD16]");
32 }

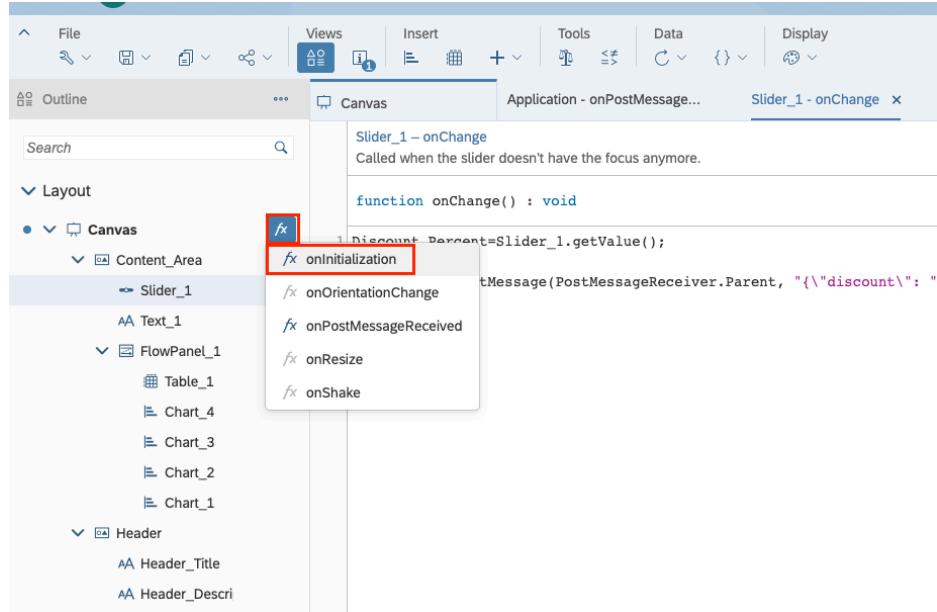
```

- 
25. Add (or copy & paste) the following scripting for the on click event of **Slider\_1**:

```
Application.postMessage(PostMessageReceiver.Parent, "{\"discount\":\n\"+ConvertUtils.numberToString(Discount_Percent) + \"}", "*");
```

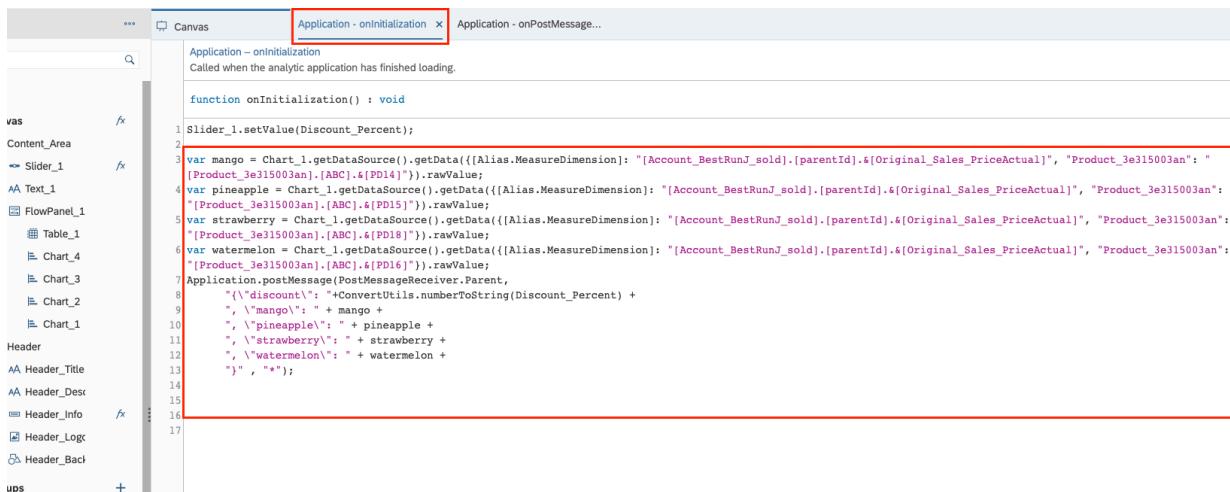


- 
26. Select **onInitialization** event of the **Canvas**

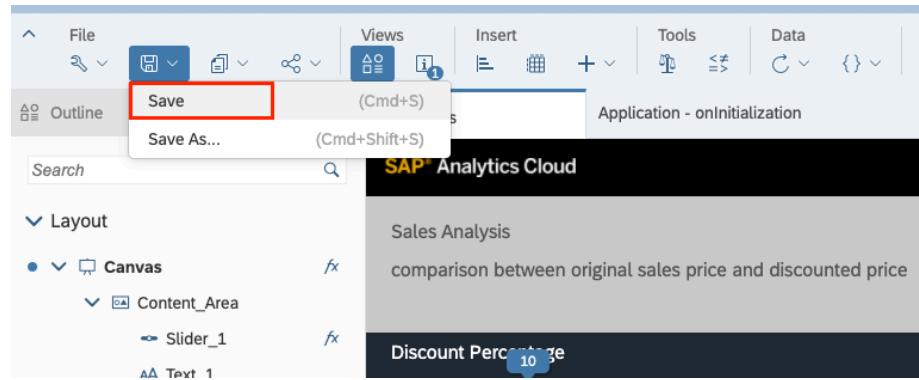


27. Add the following scripting APIs for the **onInitialization** event:

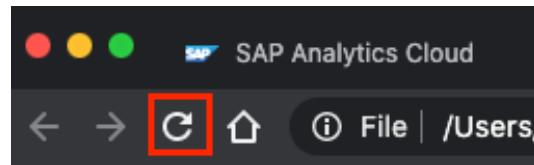
```
var mango = Chart_1.getDataSource().getData({[Alias.MeasureDimension]: "[Account_BestRunJ_sold].[parentId].[Original_Sales_PriceActual]", "Product_3e315003an": "[Product_3e315003an].[ABC].[PD14]"}).rawValue;
var pineapple = Chart_1.getDataSource().getData({[Alias.MeasureDimension]: "[Account_BestRunJ_sold].[parentId].[Original_Sales_PriceActual]", "Product_3e315003an": "[Product_3e315003an].[ABC].[PD15]"}).rawValue;
var strawberry = Chart_1.getDataSource().getData({[Alias.MeasureDimension]: "[Account_BestRunJ_sold].[parentId].[Original_Sales_PriceActual]", "Product_3e315003an": "[Product_3e315003an].[ABC].[PD18]"}).rawValue;
var watermelon = Chart_1.getDataSource().getData({[Alias.MeasureDimension]: "[Account_BestRunJ_sold].[parentId].[Original_Sales_PriceActual]", "Product_3e315003an": "[Product_3e315003an].[ABC].[PD16]"}).rawValue;
Application.postMessage(PostMessageReceiver.Parent,
  {"discount": "+ConvertUtils.numberToString(Discount_Percent) +
  ", \"mango\": " + mango +
  ", \"pineapple\": " + pineapple +
  ", \"strawberry\": " + strawberry +
  ", \"watermelon\": " + watermelon +
  "}" , "*");
```



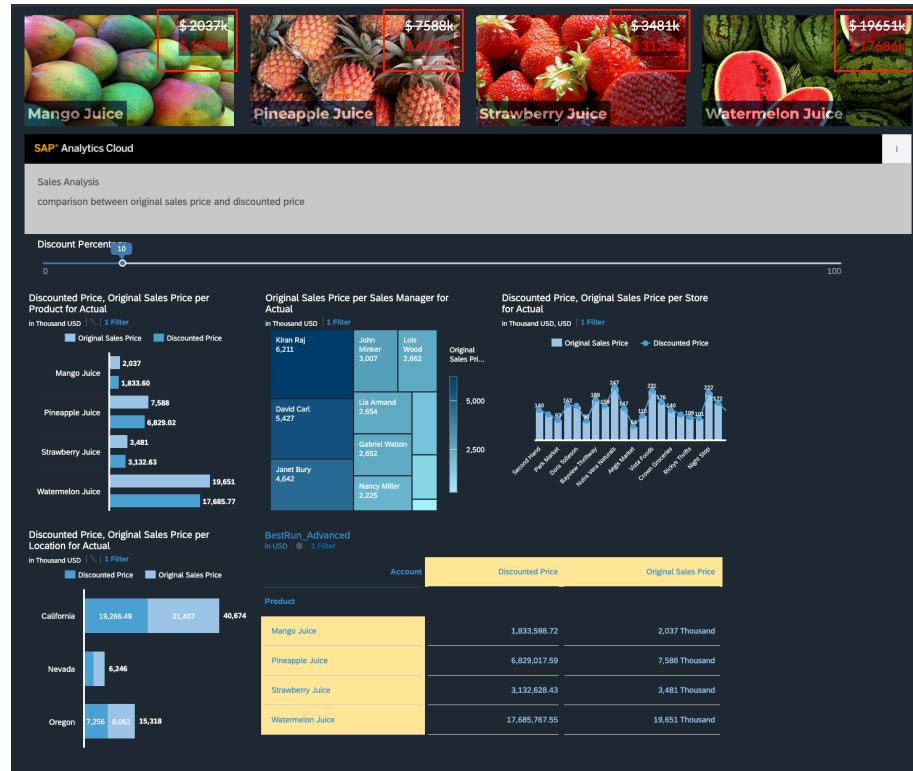
28. Click on the Floppy Disc icon and select **Save**.



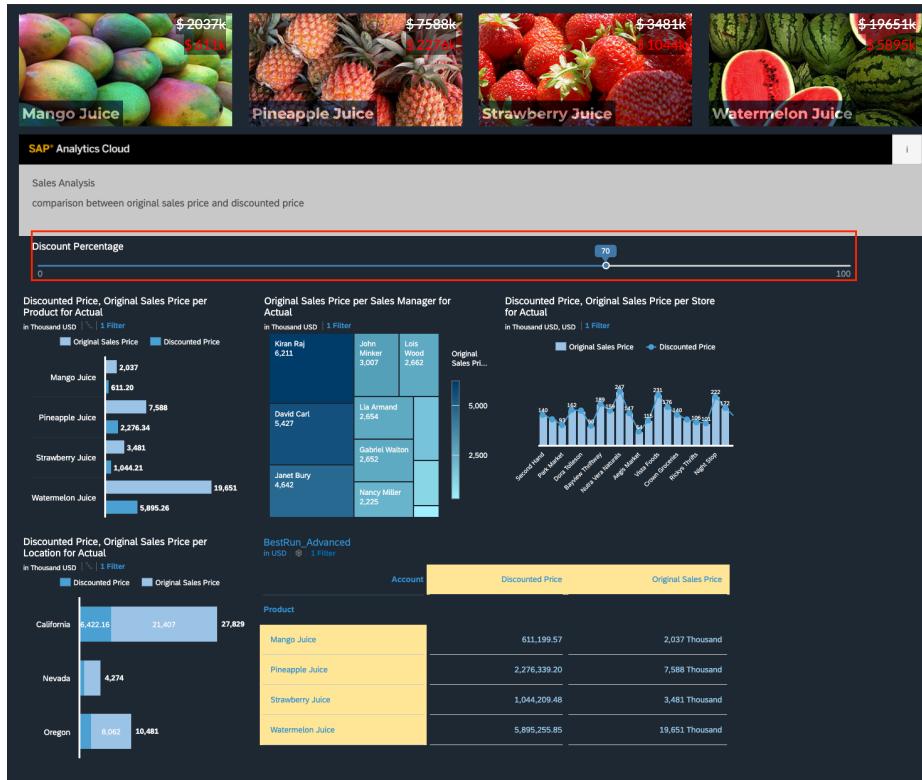
29. Click **C** on the previously opened ANA362\_EXE1.html page or enter the web address again.



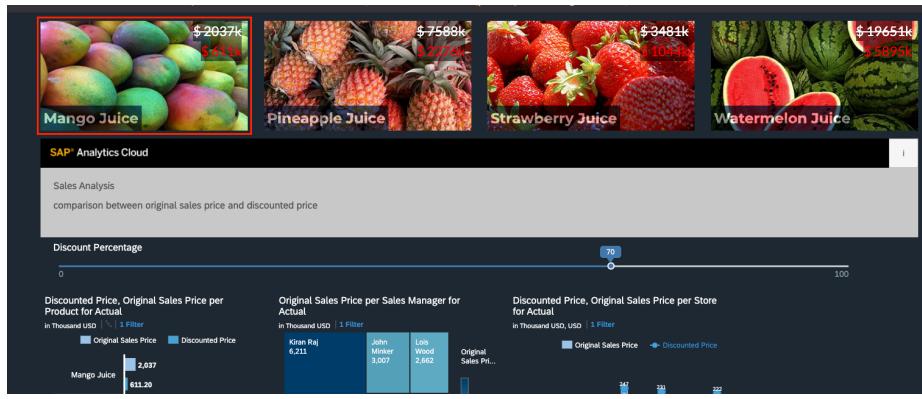
30. You can see all these prices when the application is initialized.



31. change the discount percentage to 70 via Slider

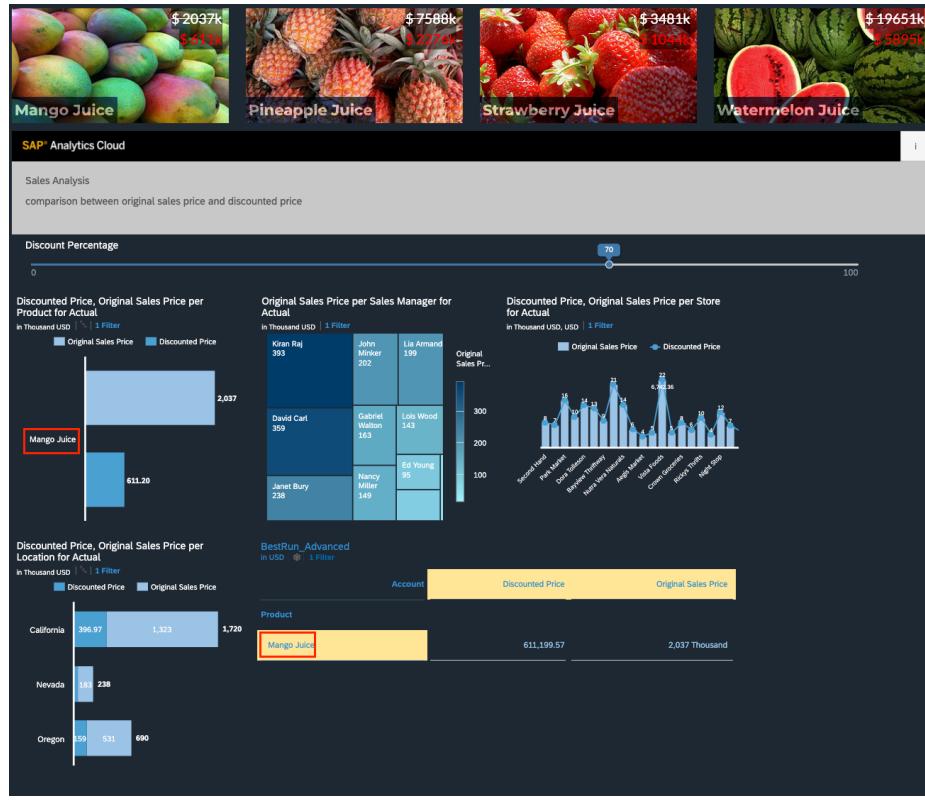


32. You can see that all the discounted prices have been changed based on the discount percentage which is **70%**.  
Here you can see the message exchange from application to HTML page.



34. Now you can see that on Mango Juice is displayed on the different Widgets. In fact, we have sent the message from HTML page “**mango juice**” to analytical application to set the filter on the Widgets. So here we can see the message exchange from **html page to analytical application**.

You can select another type of Juice to see that chart is filtered with the Juice name.



**Congratulations! You Have completed the Exercise 1.**

### ***Summary***

**You Have Successfully Completed the Exercise!**

**You are now able to:**

- Know the basic workflow of creating an Analytical Application
- Configure Table and Charts
- Understand the capability of script variable (with URL parameter)
- Get the basic concept of scripting capability
- Know how to use theme via CSS code
- Use post message to embed an analytic application into a HTML page
- Set an application to mobile enable

[www.sap.com/contactsap](http://www.sap.com/contactsap)

© 2020 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company.

The information contained herein may be changed without prior notice. Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

In particular, SAP SE or its affiliated companies have no obligation to pursue any course of business outlined in this document or any related presentation, or to develop or release any functionality mentioned therein. This document, or any related presentation, and SAP SE's or its affiliated companies' strategy and possible future developments, products, and/or platform directions and functionality are all subject to change and may be changed by SAP SE or its affiliated companies at any time for any reason without notice. The information in this document is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, and they should not be relied upon in making purchasing decisions.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies. See [www.sap.com/copyright](http://www.sap.com/copyright) for additional trademark information and notices.