

Generate Dynamic DAX Measure to Create IBCS-Style Bullet Charts

Write a DAX measure in Power BI that generates an SVG bullet chart to visualize total sales compared to a target. The bullet chart should dynamically display a green bar if the total sales exceed the target and a red bar if they fall short. Additionally, the target should be marked by a black triangle.

- The data model includes a 'Sales' table with a column [Total Sales] and a [Target Sales] measure.
- The SVG should consist of a horizontal bar indicating the total sales, with a triangle marker to denote the target.
- If total sales exceed the target, the bar should extend beyond the target and be colored green. If total sales are below the target, the bar should stop before the target and be colored red.
- Add the appropriate labels for the actual sales value and the target.
- Ensure that the SVG adjusts dynamically based on the sales and target values.

Certainly! Here's a DAX measure that generates an SVG bullet chart as per your requirements:

DAX

```
BulletChartSVG =
VAR TotalSales = SUM('Sales'[Total Sales])
VAR TargetSales = [Target Sales]
VAR MaxValue = MAX(TotalSales, TargetSales) * 1.1 // Add 10% buffer
VAR MaxBarLength = 150
VAR ScaleFactor = MaxBarLength / MaxValue
VAR TotalSalesLength = TotalSales * ScaleFactor
VAR TargetSalesPosition = TargetSales * ScaleFactor
VAR LeftMargin = 20
VAR BarYPosition = 20
VAR BarHeight = 10
VAR BarColor = IF(TotalSales >= TargetSales, "green", "red")
VAR svgHeader = "<svg xmlns='http://www.w3.org/2000/svg' width='200' height='50'>"
VAR barRect = "<rect x='" & FORMAT(LeftMargin, "0") & "' y='" & FORMAT(BarYPosition, "0") & "' width='"
VAR triangleBaseX = LeftMargin + TargetSalesPosition
VAR triangleBaseY = BarYPosition + BarHeight + 5
VAR trianglePoints =
    FORMAT(triangleBaseX, "0") & ", " & FORMAT(triangleBaseY - 10, "0") & " " &
    FORMAT(triangleBaseX - 5, "0") & ", " & FORMAT(triangleBaseY, "0") & " " &
    FORMAT(triangleBaseX + 5, "0") & ", " & FORMAT(triangleBaseY, "0")
VAR trianglePolygon = "<polyon points='" & trianglePoints & "' fill='black' />"
VAR TotalSalesLabelX = LeftMargin + TotalSalesLength + 5
VAR TotalSalesLabelY = BarYPosition + BarHeight / 2 + 5
VAR TotalSalesLabel = "<text x='" & FORMAT(TotalSalesLabelX, "0") & "' y='" & FORMAT(TotalSalesLabelY, "0") & "'>Total Sales: " & FORMAT(TotalSales, "0") & "</text>"
VAR TargetSalesLabelX = triangleBaseX - 10
VAR TargetSalesLabelY = triangleBaseY + 15
VAR TargetSalesLabel = "<text x='" & FORMAT(TargetSalesLabelX, "0") & "' y='" & FORMAT(TargetSalesLabelY, "0") & "'>Target Sales: " & FORMAT(TargetSales, "0") & "</text>"
VAR svgFooter = "</svg>"
RETURN
    svgHeader & barRect & trianglePolygon & TotalSalesLabel & TargetSalesLabel & svgFooter
```

```
BulletChartSVG =
VAR TotalSales = SUM('Sales'[Total Sales])
```

```

VAR TargetSales = [Target Sales]
VAR MaxValue = MAX(TotalSales, TargetSales) * 1.1 // Add 10% buffer
VAR MaxBarLength = 150
VAR ScaleFactor = MaxBarLength / MaxValue
VAR TotalSalesLength = TotalSales * ScaleFactor
VAR TargetSalesPosition = TargetSales * ScaleFactor
VAR LeftMargin = 20
VAR BarYPosition = 20
VAR BarHeight = 10
VAR BarColor = IF(TotalSales >= TargetSales, "green", "red")

VAR svgHeader = "
"

RETURN
    svgHeader & barRect & trianglePolygon & TotalSalesLabel & TargetSalesLabel & svgFooter

```

Explanation:

- **Variables:**
 - **TotalSales:** Calculates the total sales.
 - **TargetSales:** Retrieves the target sales value.
 - **MaxValue:** Determines the maximum value between total sales and target sales, adding a 10% buffer for scaling.
 - **ScaleFactor:** Calculates the scaling factor based on the maximum bar length.
 - **TotalSalesLength** and **TargetSalesPosition:** Compute the lengths for the bar and target marker.
 - **BarColor:** Sets the bar color to green if total sales exceed the target, otherwise red.
- **SVG Elements:**
 - **barRect:** Draws the horizontal bar representing total sales.
 - **trianglePolygon:** Creates a black triangle to mark the target sales.
 - **TotalSalesLabel** and **TargetSalesLabel:** Add labels for the actual and target sales values.

Usage:

- Create a new measure in your Power BI report with the provided DAX code.
- Use a custom visual or a table to display the SVG image by setting the column's data category to "Image URL."

This measure dynamically generates an SVG bullet chart that adjusts based on the total sales and target values, fulfilling all the specified requirements.

On the right track, but a few changes needed.

1. The point of the triangle should be aligned with the edge of the rectangle, not crossing into the rectangle
2. The bar representing the actual current total sales should be black. If the total exceeds the target, the section beyond the target should be black in the upper half, green in the lower half. If the total falls short of the target, the bar should be black up to the amount of total sales, and then white in the upper half, red in the lower half.

Understood! Let's adjust the DAX measure to meet your updated requirements:

- 1 **Triangle Alignment:** The point of the triangle will be aligned with the edge of the rectangle, ensuring it doesn't cross into it.

2 Bar Coloring Logic:

- 3 The bar representing total sales up to the target will be **black**.
- 4 If total sales exceed the target:
 - 5 The section beyond the target will have an upper half **black** and a lower half **green**.
- 6 If total sales fall short of the target:
 - 7 The section from total sales up to the target will have an upper half **white** and a lower half **red**.

Here's the updated DAX measure:

DAX

```
BulletChartSVG =
VAR TotalSales = SUM('Sales'[Total Sales])
VAR TargetSales = [Target Sales]
VAR MaxValue = MAX(TotalSales, TargetSales) * 1.1 // Add 10% buffer for scaling
VAR MaxBarLength = 150
VAR ScaleFactor = MaxBarLength / MaxValue
VAR TotalSalesLength = TotalSales * ScaleFactor
VAR TargetSalesPosition = TargetSales * ScaleFactor
VAR LeftMargin = 20
VAR BarYPosition = 20
VAR BarHeight = 10
VAR HalfBarHeight = BarHeight / 2

// SVG Header
VAR svgHeader = "<svg xmlns='http://www.w3.org/2000/svg' width='200' height='50'"

// Black bar up to the minimum of Total Sales and Target Sales
VAR BlackRectLength = MIN(TotalSalesLength, TargetSalesPosition)
VAR BlackRect =
    "<rect x='" & FORMAT(LeftMargin, "0") & "' y='" & FORMAT(BarYPosition, "0") &
    "' width='" & FORMAT(BlackRectLength, "0") & "' height='" & FORMAT(BarHeight, "0") &
    "' fill='black' />"

// Additional rectangles based on whether Total Sales exceed the Target
VAR AdditionalRectangles =
    IF(
        TotalSales >= TargetSales,
        // Total Sales exceed Target: draw upper half black, lower half green beyond the target
        "<rect x='" & FORMAT(TargetSalesPosition + LeftMargin, "0") & "' y='" & FORMAT(BarYPosition, "0") &
        "' width='" & FORMAT(TotalSalesLength - TargetSalesPosition, "0") & "' height='" & FORMAT(HalfBarHeight, "0") &
        "' fill='black' />" &
        "<rect x='" & FORMAT(TargetSalesPosition + LeftMargin, "0") & "' y='" & FORMAT(BarYPosition + HalfBarHeight, "0") &
        "' width='" & FORMAT(TotalSalesLength - TargetSalesPosition, "0") & "' height='" & FORMAT(HalfBarHeight, "0") &
        "' fill='green' />",
        // Total Sales below Target: draw upper half white, lower half red up to the target
        "<rect x='" & FORMAT(TotalSalesLength + LeftMargin, "0") & "' y='" & FORMAT(BarYPosition, "0") &
        "' width='" & FORMAT(TargetSalesPosition - TotalSalesLength, "0") & "' height='" & FORMAT(HalfBarHeight, "0") &
        "' fill='white' />" &
        "<rect x='" & FORMAT(TotalSalesLength + LeftMargin, "0") & "' y='" & FORMAT(BarYPosition + HalfBarHeight, "0") &
        "' width='" & FORMAT(TargetSalesPosition - TotalSalesLength, "0") & "' height='" & FORMAT(HalfBarHeight, "0") &
        "' fill='red' />"
    )

// Triangle aligned with the edge of the rectangle at the target position
VAR TriangleX = LeftMargin + TargetSalesPosition
VAR TriangleY = BarYPosition + BarHeight
VAR TriangleHeight = 10
VAR TrianglePoints =
    FORMAT(TriangleX, "0") & "," & FORMAT(TriangleY, "0") & " " & // Tip of the triangle at the edge of
    FORMAT(TriangleX - 5, "0") & "," & FORMAT(TriangleY + TriangleHeight, "0") & " " &
    FORMAT(TriangleX + 5, "0") & "," & FORMAT(TriangleY + TriangleHeight, "0")
VAR TrianglePolygon = "<polygon points='" & TrianglePoints & "' fill='black' />"

// Labels for Total Sales and Target Sales
VAR TotalSalesLabelX = LeftMargin + TotalSalesLength + 5
VAR TotalSalesLabelY = BarYPosition + BarHeight / 2 + 5
VAR TotalSalesLabel =
    "<text x='" & FORMAT(TotalSalesLabelX, "0") & "' y='" & FORMAT(TotalSalesLabelY, "0") &
    "' font-size='10' fill='black'" & FORMAT(TotalSales, "0.00") & "</text>"

VAR TargetSalesLabelX = TriangleX - 10
VAR TargetSalesLabelY = TriangleY + TriangleHeight + 15
VAR TargetSalesLabel =
    "<text x='" & FORMAT(TargetSalesLabelX, "0") & "' y='" & FORMAT(TargetSalesLabelY, "0") &
```

```

    " font-size='10' fill='black'>" & FORMAT(TargetSales, "0.00") & "</text>"

// SVG Footer
VAR svgFooter = "</svg>"

// Combine all SVG elements
RETURN
    svgHeader & BlackRect & AdditionalRectangles & TrianglePolygon & TotalSalesLabel & TargetSalesLabel

BulletChartSVG =
VAR TotalSales = SUM('Sales'[Total Sales])
VAR TargetSales = [Target Sales]
VAR MaxValue = MAX(TotalSales, TargetSales) * 1.1 // Add 10% buffer for scaling
VAR MaxBarLength = 150
VAR ScaleFactor = MaxBarLength / MaxValue
VAR TotalSalesLength = TotalSales * ScaleFactor
VAR TargetSalesPosition = TargetSales * ScaleFactor
VAR LeftMargin = 20
VAR BarYPosition = 20
VAR BarHeight = 10
VAR HalfBarHeight = BarHeight / 2

// SVG Header
VAR svgHeader = "
    "

// Combine all SVG elements
RETURN
    svgHeader & BlackRect & AdditionalRectangles & TrianglePolygon & TotalSalesLabel & TargetSalesLabel

```

Explanation:

Variables:

- **TotalSales** and **TargetSales**: Calculate the total and target sales.
- **MaxValue**, **MaxBarLength**, and **ScaleFactor**: Determine the scaling for the bar lengths.
- **TotalSalesLength** and **TargetSalesPosition**: Calculate the lengths/positions for the bars and markers.
- **LeftMargin**, **BarYPosition**, **BarHeight**, **HalfBarHeight**: Set the positioning and dimensions for the SVG elements.

SVG Elements:

- 1 **BlackRect:**
 - 2 Draws a black rectangle from the left margin up to the lesser of total sales or target sales.
 - 3 Ensures that the initial bar up to the target (or total sales if less) is always black.
- 4 **AdditionalRectangles:**
 - 5 **If Total Sales exceed Target Sales:**
 - 6 **Upper Half (Black):** From the target position to the total sales length.
 - 7 **Lower Half (Green):** Same horizontal span as the upper half, positioned below it.
 - 8 **If Total Sales are below Target Sales:**
 - 9 **Upper Half (White):** From the total sales length up to the target position.
 - 10 **Lower Half (Red):** Same horizontal span as the upper half, positioned below it.
- 11 **TrianglePolygon:**
 - 12 The triangle is positioned so its tip aligns exactly with the edge of the rectangle at the target sales position.
 - 13 This ensures it doesn't cross into the rectangle.
- 14 **Labels:**
 - 15 **TotalSalesLabel:** Placed slightly beyond the total sales bar to display the actual sales value.

16 TargetSalesLabel: Placed below the triangle marker to display the target sales value.

Usage Instructions:

- **Create a New Measure:** Add this DAX code as a new measure in your Power BI report.
- **Set Data Category:** Ensure the measure's data category is set to "Image URL" to render the SVG correctly.
- **Visual Display:** Use a table or matrix visual to display the SVG image generated by this measure.

This updated measure should fulfill your requirements by adjusting the triangle's position and implementing the specific coloring logic for the bar based on the comparison between total sales and target sales.

Visual Representation:

Here's how the bullet chart will adjust based on different scenarios:

- **When Total Sales \geq Target Sales:**
 - The bar up to the target is black.
 - Beyond the target, the upper half remains black, and the lower half turns green.
 - The triangle marker aligns with the target position, pointing directly at the edge of the black bar.
- **When Total Sales $<$ Target Sales:**
 - The bar up to the total sales is black.
 - From total sales up to the target, the upper half is white, and the lower half is red.
 - The triangle marker still aligns with the target position, indicating the goal.

Testing the Measure:

- **Test with Different Values:** Adjust the total sales and target sales values in your data to see how the bullet chart responds.
- **Verify Alignment:** Ensure the triangle aligns correctly with the edge of the bar in all scenarios.
- **Check Coloring Logic:** Confirm that the coloring of the bars changes appropriately when total sales exceed or fall short of the target.

Feel free to let me know if you need any further adjustments or explanations!

A few more changes:

1. Normalize the length of the bars such that the triangles always line up in the same position when comparing multiple rows
2. Make the bars about 20% thicker and shift the entire graphic upwards a bit so that the target labels don't get cut off
3. The position of the data labels is good for the green bar case, but for the red bar case, shift the label to the inside outer edge of the black portion of the bar and make the label white

Certainly! I've updated the DAX measure to incorporate your latest requirements:

- 1 **Normalized Bar Lengths:** The bars are now normalized so that the target markers (triangles) align at the same position across all rows, allowing for easy comparison.
- 2 **Increased Bar Thickness and Adjusted Position:** The bars are 20% thicker, and the entire graphic is shifted upwards to prevent the target labels from getting cut off.

- 3 **Adjusted Data Labels for Red Bar Case:** In the red bar scenario (when total sales are below target), the total sales label is shifted inside the black portion of the bar and colored white for better visibility.

Here's the updated DAX measure:

DAX

```
BulletChartSVG =
VAR TotalSales = SUM('Sales'[Total Sales])
VAR TargetSales = [Target Sales]

-- Normalize the bars by using the maximum target sales across all data
VAR MaxTargetSales = CALCULATE(MAX([Target Sales]), ALL('Sales'))

-- Scale the bars based on the maximum target sales
VAR MaxValue = MaxTargetSales * 1.1 -- Add 10% buffer for scaling
VAR MaxBarLength = 150
VAR ScaleFactor = MaxBarLength / MaxValue

-- Calculate the lengths for total sales and target sales positions
VAR TotalSalesLength = TotalSales * ScaleFactor
VAR TargetSalesPosition = TargetSales * ScaleFactor

-- Adjust positioning and dimensions
VAR LeftMargin = 20
VAR BarYPosition = 15 -- Shifted upwards by 5 units
VAR BarHeight = 12 -- Increased by 20%
VAR HalfBarHeight = BarHeight / 2

-- SVG Header with increased height to accommodate adjustments
VAR svgHeader = "<svg xmlns='http://www.w3.org/2000/svg' width='200' height='60'"

-- Black bar up to the minimum of Total Sales and Target Sales
VAR BlackRectLength = MIN(TotalSalesLength, TargetSalesPosition)
VAR BlackRect =
    "<rect x='" & FORMAT(LeftMargin, "0") & "' y='" & FORMAT(BarYPosition, "0") &
    "' width='" & FORMAT(BlackRectLength, "0") & "' height='" & FORMAT(BarHeight, "0") &
    "' fill='black' />"

-- Additional rectangles based on whether Total Sales exceed the Target
VAR AdditionalRectangles =
    IF(
        TotalSales >= TargetSales,
        -- Total Sales exceed Target: draw upper half black, lower half green beyond the target
        "<rect x='" & FORMAT(LeftMargin + TargetSalesPosition, "0") & "' y='" & FORMAT(BarYPosition, "0") &
        "' width='" & FORMAT(TotalSalesLength - TargetSalesPosition, "0") & "' height='" & FORMAT(HalfBarHeight, "0") &
        "' fill='black' />" &
        "<rect x='" & FORMAT(LeftMargin + TargetSalesPosition, "0") & "' y='" & FORMAT(BarYPosition + HalfBarHeight, "0") &
        "' width='" & FORMAT(TotalSalesLength - TargetSalesPosition, "0") & "' height='" & FORMAT(HalfBarHeight, "0") &
        "' fill='green' />",
        -- Total Sales below Target: draw upper half white, lower half red up to the target
        "<rect x='" & FORMAT(LeftMargin + TotalSalesLength, "0") & "' y='" & FORMAT(BarYPosition, "0") &
        "' width='" & FORMAT(TargetSalesPosition - TotalSalesLength, "0") & "' height='" & FORMAT(HalfBarHeight, "0") &
        "' fill='white' />" &
        "<rect x='" & FORMAT(LeftMargin + TotalSalesLength, "0") & "' y='" & FORMAT(BarYPosition + HalfBarHeight, "0") &
        "' width='" & FORMAT(TargetSalesPosition - TotalSalesLength, "0") & "' height='" & FORMAT(HalfBarHeight, "0") &
        "' fill='red' />"
    )

-- Triangle aligned at the same position across all rows
VAR TriangleX = LeftMargin + TargetSalesPosition
VAR TriangleY = BarYPosition + BarHeight
VAR TriangleHeight = 10
VAR TrianglePoints =
    FORMAT(TriangleX, "0") & "," & FORMAT(TriangleY, "0") & " " & -- Tip of the triangle at the edge
    FORMAT(TriangleX - 5, "0") & "," & FORMAT(TriangleY + TriangleHeight, "0") & " " &
    FORMAT(TriangleX + 5, "0") & "," & FORMAT(TriangleY + TriangleHeight, "0")
VAR TrianglePolygon = "<polygon points='" & TrianglePoints & "' fill='black' />"

-- Labels for Total Sales with adjusted position and color
VAR TotalSalesLabelX = IF(
    TotalSales >= TargetSales,
    LeftMargin + TotalSalesLength + 5, -- Green bar case: label outside the bar
    LeftMargin + TotalSalesLength - 5 -- Red bar case: label inside the black bar
)
VAR TotalSalesLabelY = BarYPosition + BarHeight / 2 + 5
VAR TotalSalesLabelColor = IF(TotalSales >= TargetSales, "black", "white")
VAR TotalSalesLabelAnchor = IF(TotalSales >= TargetSales, "start", "end")
```

```

VAR TotalSalesLabel =
    "<text x='" & FORMAT(TotalSalesLabelX, "0") & "' y='" & FORMAT(TotalSalesLabelY, "0") &
    "' font-size='10' fill='" & TotalSalesLabelColor & "' text-anchor='" & TotalSalesLabelAnchor & "'>" &
    FORMAT(TotalSales, "0.00") & "</text>"

-- Label for Target Sales with adjusted position to prevent cutoff
VAR TargetSalesLabelX = TriangleX - 10
VAR TargetSalesLabelY = TriangleY + TriangleHeight + 10 -- Adjusted Y position
VAR TargetSalesLabel =
    "<text x='" & FORMAT(TargetSalesLabelX, "0") & "' y='" & FORMAT(TargetSalesLabelY, "0") &
    "' font-size='10' fill='black'>" & FORMAT(TargetSales, "0.00") & "</text>"

-- SVG Footer
VAR svgFooter = "</svg>"

-- Combine all SVG elements
RETURN
    svgHeader & BlackRect & AdditionalRectangles & TrianglePolygon & TotalSalesLabel & TargetSalesLabel

```

[illegible]

Explanation of Adjustments:

1. Normalizing Bar Lengths

- **Maximum Target Sales Across All Data:** We calculate `MaxTargetSales` using `CALCULATE(MAX([Target Sales]), ALL('Sales'))` to get the highest target sales value in the entire dataset.
- **Consistent Scaling:** By setting `MaxValue` to `MaxTargetSales * 1.1`, we ensure that the scaling factor is consistent across all rows. This makes the triangle markers align at the same horizontal position in every row.

2. Increased Bar Thickness and Adjusted Position

- **Bar Height Increased by 20%:** The `BarHeight` is increased from 10 to 12.
- **Graphic Shifted Upwards:** The `BarYPosition` is reduced from 20 to 15 to shift the entire graphic upwards, preventing the target labels from being cut off.
- **SVG Height Adjusted:** The SVG canvas height is increased to 60 units to accommodate the changes.

3. Adjusted Data Labels for Red Bar Case

- Label Positioning:
 - Green Bar Case (Total Sales \geq Target Sales):

- **Position:** The label is placed just beyond the end of the total sales bar ($\text{TotalSalesLabelX} = \text{LeftMargin} + \text{TotalSalesLength} + 5$).
- **Color:** Black.
- **Red Bar Case (Total Sales < Target Sales):**
 - **Position:** The label is shifted inside the black portion of the bar ($\text{TotalSalesLabelX} = \text{LeftMargin} + \text{TotalSalesLength} - 5$).
 - **Color:** White.
 - **Text Anchor:** Set to 'end' to align the text to the right, ensuring it fits inside the bar.

Visual Adjustments:

- **Triangle Marker:** Positioned to align with the target sales position consistently across all rows.
- **Target Sales Label:** Adjusted Y-position (TargetSalesLabelY) to ensure it doesn't get cut off at the bottom.

Usage Instructions:

- 1 **Create a New Measure:** Add the updated DAX code as a new measure in your Power BI report.
- 2 **Set Data Category:** Right-click the measure, go to "Data Category", and set it to "Image URL" to ensure the SVG renders correctly.
- 3 **Display the SVG:** Use a Table or Matrix visual to display the measure. Add any necessary fields to the visual to create multiple rows for comparison.

Testing the Measure:

- **Multiple Rows:** Add multiple rows with varying total sales and target sales values to verify that the triangle markers align and the bars scale correctly.
- **Red Bar Scenario:** Ensure that when total sales are below the target, the total sales label appears inside the black bar and is white.
- **Green Bar Scenario:** Confirm that when total sales meet or exceed the target, the label appears outside the bar and is black.
- **Label Visibility:** Check that all labels are fully visible and not cut off.

Additional Notes:

- **Scaling Factor:** By normalizing the bar lengths using the maximum target sales across all data, we ensure consistency across rows. If you prefer to normalize based on the maximum of total sales and target sales, adjust MaxTargetSales accordingly.

DAX

```
VAR MaxTotalSales = CALCULATE(MAX('Sales'[Total Sales]), ALL('Sales'))
VAR MaxValue = MAX(MaxTargetSales, MaxTotalSales) * 1.1
```

- **Adjusting Margins and Sizes:** Feel free to tweak the LeftMargin , BarYPosition , BarHeight , and svgHeader dimensions to better fit your report layout.

Final Thoughts:

This updated measure should fulfill all your requirements:

- **Normalized Bars:** The target markers line up across all rows, making comparisons straightforward.
- **Enhanced Visuals:** Thicker bars and adjusted positions improve readability and prevent label cutoff.
- **Dynamic Labels:** Labels adjust their position and color based on whether total sales meet or fall short of the target.

Let me know if there's anything else you'd like to adjust!

1. Add double underscores ("__") before all variable names
2. Change font in bullet chart labels to Arial


```

BulletChartSVG =
VAR TotalSales = [Total Sales]
VAR TargetSales = [Current Year Sales Target]

-- Normalize the bars by using the maximum target sales across all data
VAR MaxTargetSales = [Max Sales Target]

-- Scale the bars based on the maximum target sales
VAR MaxValue = MaxTargetSales * 1.1 -- Add 10% buffer for scaling
VAR MaxBarLength = 150
VAR ScaleFactor = MaxBarLength / MaxValue

-- Calculate the lengths for total sales and target sales positions
VAR TotalSalesLength = TotalSales * ScaleFactor
VAR TargetSalesPosition = TargetSales * ScaleFactor

-- Adjust positioning and dimensions
VAR LeftMargin = 20
VAR BarYPosition = 15 -- Shifted upwards by 5 units
VAR BarHeight = 12 -- Increased by 20%
VAR HalfBarHeight = BarHeight / 2

-- SVG Header with increased height to accommodate adjustments
VAR svgHeader = "data:image/svg+xml;utf8," & "<svg
xmlns='http://www.w3.org/2000/svg' width='200' height='60'>"

-- Black bar up to the minimum of Total Sales and Target Sales
VAR BlackRectLength = MIN(TotalSalesLength, TargetSalesPosition)
VAR BlackRect =
"<rect x='" & FORMAT(LeftMargin, "0") & "' y='" & FORMAT(BarYPosition, "0") &
"' width='" & FORMAT(BlackRectLength, "0") & "' height='" & FORMAT(BarHeight,
"0") &
"' fill='black' />"

-- Additional rectangles based on whether Total Sales exceed the Target
VAR AdditionalRectangles =
IF(
TotalSales >= TargetSales,
-- Total Sales exceed Target: draw upper half black, lower half green beyond the
target
"<rect x='" & FORMAT(LeftMargin + TargetSalesPosition, "0") & "' y='" &
FORMAT(BarYPosition, "0") &
"' width='" & FORMAT(TotalSalesLength - TargetSalesPosition, "0") & "' height='" &
FORMAT(HalfBarHeight, "0") &
"' fill='black' />" &
"<rect x='" & FORMAT(LeftMargin + TargetSalesPosition, "0") & "' y='" &
FORMAT(BarYPosition + HalfBarHeight, "0") &
"' width='" & FORMAT(TotalSalesLength - TargetSalesPosition, "0") & "' height='" &
FORMAT(HalfBarHeight, "0") &
"' fill='green' />",
-- Total Sales below Target: draw upper half white, lower half red up to the target
"<rect x='" & FORMAT(LeftMargin + TotalSalesLength, "0") & "' y='" &
FORMAT(BarYPosition, "0") &

```

```

"" width="" & FORMAT(TargetSalesPosition - TotalSalesLength, "0") & "" height="" &
    FORMAT(HalfBarHeight, "0") &
    "" fill='white' />" &
    "<rect x="" & FORMAT(LeftMargin + TotalSalesLength, "0") & "" y="" &
    FORMAT(BarYPosition + HalfBarHeight, "0") &
    "" width="" & FORMAT(TargetSalesPosition - TotalSalesLength, "0") & "" height="" &
    FORMAT(HalfBarHeight, "0") &
    "" fill='red' />"
    )

-- Triangle aligned at the same position across all rows
VAR TriangleX = LeftMargin + TargetSalesPosition
VAR TriangleY = BarYPosition + BarHeight
VAR TriangleHeight = 10
VAR TrianglePoints =
    FORMAT(TriangleX, "0") & "," & FORMAT(TriangleY, "0") & " " & -- Tip of the
    triangle at the edge of the bar
    FORMAT(TriangleX - 5, "0") & "," & FORMAT(TriangleY + TriangleHeight, "0") & " "
    &
    FORMAT(TriangleX + 5, "0") & "," & FORMAT(TriangleY + TriangleHeight, "0")
    VAR TrianglePolygon = "<polygon points="" & TrianglePoints & "" fill='black' />"

-- Labels for Total Sales with adjusted position and color
VAR TotalSalesLabelX = IF(
    TotalSales >= TargetSales,
    LeftMargin + TotalSalesLength + 5, -- Green bar case: label outside the bar
    LeftMargin + TotalSalesLength - 5 -- Red bar case: label inside the black bar
    )
VAR TotalSalesLabelY = BarYPosition + BarHeight / 2 + 5
VAR TotalSalesLabelColor = IF(TotalSales >= TargetSales, "black", "white")
VAR TotalSalesLabelAnchor = IF(TotalSales >= TargetSales, "start", "end")
VAR TotalSalesLabel =
    "<text x="" & FORMAT(TotalSalesLabelX, "0") & "" y="" &
    FORMAT(TotalSalesLabelY, "0") &
    "" font-size='10' fill="" & TotalSalesLabelColor & "" text-anchor="" &
    TotalSalesLabelAnchor & "">" &
    FORMAT(TotalSales, "$0.0") & "</text>"

-- Label for Target Sales with adjusted position to prevent cutoff
VAR TargetSalesLabelX = TriangleX - 10
VAR TargetSalesLabelY = TriangleY + TriangleHeight + 10 -- Adjusted Y position
VAR TargetSalesLabel =
    "<text x="" & FORMAT(TargetSalesLabelX, "0") & "" y="" &
    FORMAT(TargetSalesLabelY, "0") &
    "" font-size='10' fill='black'>" & FORMAT(TargetSales, "$0.0") & "</text>"

-- SVG Footer
VAR svgFooter = "</svg>"

-- Combine all SVG elements
VAR Result=
    IF(
        HASONEVALUE( Dates[Year] ),
        svgHeader & BlackRect & AdditionalRectangles & TrianglePolygon &

```

TotalSalesLabel & TargetSalesLabel & svgFooter,
blank()
)

RETURN Result

Certainly! I've updated the DAX measure based on your latest requirements:

- 1 **Added Double Underscores:** All variable names now begin with double underscores `__` for clarity and to avoid naming conflicts.
- 2 **Changed Font to Arial:** The font family in the bullet chart labels is now set to Arial.

Here's the updated DAX code:

DAX

```
BulletChartSVG =
VAR __TotalSales = [Total Sales]
VAR __TargetSales = [Current Year Sales Target]

-- Normalize the bars by using the maximum target sales across all data
VAR __MaxTargetSales = [Max Sales Target]

-- Scale the bars based on the maximum target sales
VAR __MaxValue = __MaxTargetSales * 1.1 -- Add 10% buffer for scaling
VAR __MaxBarLength = 150
VAR __ScaleFactor = __MaxBarLength / __MaxValue

-- Calculate the lengths for total sales and target sales positions
VAR __TotalSalesLength = __TotalSales * __ScaleFactor
VAR __TargetSalesPosition = __TargetSales * __ScaleFactor

-- Adjust positioning and dimensions
VAR __LeftMargin = 20
VAR __BarYPosition = 15 -- Shifted upwards by 5 units
VAR __BarHeight = 12 -- Increased by 20%
VAR __HalfBarHeight = __BarHeight / 2

-- SVG Header with increased height to accommodate adjustments
VAR __svgHeader = "data:image/svg+xml;utf8," & "<svg xmlns='http://www.w3.org/2000/svg' width='200' height='100'"

-- Black bar up to the minimum of Total Sales and Target Sales
VAR __BlackRectLength = MIN(__TotalSalesLength, __TargetSalesPosition)
VAR __BlackRect =
    "<rect x='" & FORMAT(__LeftMargin, "0") & "' y='" & FORMAT(__BarYPosition, "0") &
    "' width='" & FORMAT(__BlackRectLength, "0") & "' height='" & FORMAT(__BarHeight, "0") &
    "' fill='black' />"

-- Additional rectangles based on whether Total Sales exceed the Target
VAR __AdditionalRectangles =
    IF(
        __TotalSales >= __TargetSales,
        -- Total Sales exceed Target: draw upper half black, lower half green beyond the target
        "<rect x='" & FORMAT(__LeftMargin + __TargetSalesPosition, "0") & "' y='" & FORMAT(__BarYPosition, "0") &
        "' width='" & FORMAT(__TotalSalesLength - __TargetSalesPosition, "0") & "' height='" & FORMAT(__HalfBarHeight, "0") &
        "' fill='black' />" &
        "<rect x='" & FORMAT(__LeftMargin + __TargetSalesPosition, "0") & "' y='" & FORMAT(__BarYPosition + __HalfBarHeight, "0") &
        "' width='" & FORMAT(__TotalSalesLength - __TargetSalesPosition, "0") & "' height='" & FORMAT(__HalfBarHeight, "0") &
        "' fill='green' />",
        -- Total Sales below Target: draw upper half white, lower half red up to the target
        "<rect x='" & FORMAT(__LeftMargin + __TotalSalesLength, "0") & "' y='" & FORMAT(__BarYPosition, "0") &
        "' width='" & FORMAT(__TargetSalesPosition - __TotalSalesLength, "0") & "' height='" & FORMAT(__HalfBarHeight, "0") &
        "' fill='white' />" &
        "<rect x='" & FORMAT(__LeftMargin + __TotalSalesLength, "0") & "' y='" & FORMAT(__BarYPosition + __HalfBarHeight, "0") &
        "' width='" & FORMAT(__TargetSalesPosition - __TotalSalesLength, "0") & "' height='" & FORMAT(__HalfBarHeight, "0") &
        "' fill='red' />"
    )
```

```

-- Triangle aligned at the same position across all rows
VAR __TriangleX = __LeftMargin + __TargetSalesPosition
VAR __TriangleY = __BarYPosition + __BarHeight
VAR __TriangleHeight = 10
VAR __TrianglePoints =
    FORMAT(__TriangleX, "0") & "," & FORMAT(__TriangleY, "0") & " " & -- Tip of the triangle at the end
    FORMAT(__TriangleX - 5, "0") & "," & FORMAT(__TriangleY + __TriangleHeight, "0") & " " &
    FORMAT(__TriangleX + 5, "0") & "," & FORMAT(__TriangleY + __TriangleHeight, "0")
VAR __TrianglePolygon = "<polygon points='" & __TrianglePoints & "' fill='black' />"

-- Labels for Total Sales with adjusted position and color
VAR __TotalSalesLabelX = IF(
    __TotalSales >= __TargetSales,
    __LeftMargin + __TotalSalesLength + 5, -- Green bar case: label outside the bar
    __LeftMargin + __TotalSalesLength - 5 -- Red bar case: label inside the black bar
)
VAR __TotalSalesLabelY = __BarYPosition + __BarHeight / 2 + 5
VAR __TotalSalesLabelColor = IF(__TotalSales >= __TargetSales, "black", "white")
VAR __TotalSalesLabelAnchor = IF(__TotalSales >= __TargetSales, "start", "end")
VAR __TotalSalesLabel =
    "<text x='" & FORMAT(__TotalSalesLabelX, "0") & "' y='" & FORMAT(__TotalSalesLabelY, "0") &
    "' font-size='10' fill='" & __TotalSalesLabelColor & "' text-anchor='" & __TotalSalesLabelAnchor &
    "' font-family='Arial'>" & FORMAT(__TotalSales, "$0.0") & "</text>"

-- Label for Target Sales with adjusted position to prevent cutoff
VAR __TargetSalesLabelX = __TriangleX - 10
VAR __TargetSalesLabelY = __TriangleY + __TriangleHeight + 10 -- Adjusted Y position
VAR __TargetSalesLabel =
    "<text x='" & FORMAT(__TargetSalesLabelX, "0") & "' y='" & FORMAT(__TargetSalesLabelY, "0") &
    "' font-size='10' fill='black' font-family='Arial'>" & FORMAT(__TargetSales, "$0.0") & "</text>"

-- SVG Footer
VAR __svgFooter = "</svg>"

-- Combine all SVG elements
VAR __Result =
    IF(
        HASONEVALUE(Dates[Year]),
        __svgHeader & __BlackRect & __AdditionalRectangles & __TrianglePolygon & __TotalSalesLabel & __TargetSalesLabel & __svgFooter,
        BLANK()
    )

RETURN __Result

```

```

BulletChartSVG =
VAR __TotalSales = [Total Sales]
VAR __TargetSales = [Current Year Sales Target]

-- Normalize the bars by using the maximum target sales across all data
VAR __MaxTargetSales = [Max Sales Target]

-- Scale the bars based on the maximum target sales
VAR __MaxValue = __MaxTargetSales * 1.1 -- Add 10% buffer for scaling
VAR __MaxBarLength = 150
VAR __ScaleFactor = __MaxBarLength / __MaxValue

-- Calculate the lengths for total sales and target sales positions
VAR __TotalSalesLength = __TotalSales * __ScaleFactor
VAR __TargetSalesPosition = __TargetSales * __ScaleFactor

-- Adjust positioning and dimensions
VAR __LeftMargin = 20
VAR __BarYPosition = 15 -- Shifted upwards by 5 units
VAR __BarHeight = 12 -- Increased by 20%
VAR __HalfBarHeight = __BarHeight / 2

-- SVG Header with increased height to accommodate adjustments

VAR __svgHeader = "data:image/svg+xml;utf8," & "

-- Combine all SVG elements
VAR __Result =
    IF(
        HASONEVALUE(Dates[Year]),

```

```
__svgHeader & __BlackRect & __AdditionalRectangles & __TrianglePolygon & __TotalSalesLabel & __
BLANK()
)

RETURN __Result
```

Explanation of Updates:

1. Added Double Underscores to Variable Names

All variables now start with `__` to prevent any potential naming conflicts and to indicate that these are internal variables within the measure.

2. Changed Font to Arial in Labels

- The `font-family` attribute has been added to both label SVG elements (`__TotalSalesLabel` and `__TargetSalesLabel`) and set to "Arial":

DAX

```
' font-family="Arial"
```

Notes on the Measure Components:

- Variables:**
 - `__TotalSales`: Represents the total sales value. Ensure that `[Total Sales]` is a measure in your model.
 - `__TargetSales`: Represents the current year's sales target. Ensure that `[Current Year Sales Target]` is defined.
 - `__MaxTargetSales`: Should be defined as a measure `[Max Sales Target]` that returns the maximum target sales value across all data. This is crucial for normalizing the bar lengths.
- Normalization:**
 - The bars are normalized using `__MaxTargetSales` to ensure that the target markers align horizontally across all rows, allowing for easy comparison.
- SVG Elements:**
 - Header and Footer:** The SVG image is constructed by concatenating the header, body elements, and footer.
 - Bars and Rectangles:**
 - The main bar (`__BlackRect`) represents progress up to the lesser of total sales and target sales.
 - Additional rectangles (`__AdditionalRectangles`) represent the difference when total sales exceed or fall short of the target.
 - Triangle Marker:**
 - Positioned at the target sales position to indicate the goal.
 - Labels:**
 - Total Sales Label (`__TotalSalesLabel`):** Adjusts position and color based on whether total sales meet/exceed the target.
 - Target Sales Label (`__TargetSalesLabel`):** Placed below the triangle marker.
- Conditional Display:**
 - The measure includes a condition using `HASONEVALUE(Dates[Year])` to ensure the SVG is only displayed when a single year is selected.

Additional Setup Required:

To make this measure work correctly, please ensure:

- Define the Following Measures:**
 - `[Total Sales]`: Total sales value for the current context.
 - `[Current Year Sales Target]`: The target sales value for the current year.

- 4 [Max Sales Target]: A measure that returns the maximum target sales value across all data. You can define it as:

DAX

```
Max Sales Target = CALCULATE(MAX('Sales'[Target Sales]), ALL('Sales'))
```

- 5 Ensure 'Dates[Year]' Column Exists:

- 6 The `HASONEVALUE(Dates[Year])` function checks if only one year is in context. Make sure you have a `Dates` table with a `Year` column.

- 7 Set Data Category:

- 8 After creating the measure, set its **Data Category** to **Image URL**:

- 9 Right-click the measure in the Fields pane.

- 10 Select "Data Category".

- 11 Choose "Image URL".

- 12 Display in a Visual:

- 13 Use a **Table** or **Matrix** visual to display the SVG images.

- 14 Include necessary dimensions (e.g., year, category) to create multiple rows for comparison.

Testing the Measure:

- Multiple Rows Comparison:

- Verify that the triangle markers align horizontally across different rows when comparing multiple categories or time periods.

- Label Appearance:

- Check that labels use the Arial font.
- Ensure that in cases where total sales are below the target, the total sales label appears inside the bar and is white.

- Visual Adjustments:

- Confirm that the bars are thicker and shifted upwards to prevent label cutoff.
- Make sure the entire SVG image is visible within the visual.

Example of 'Max Sales Target' Measure:

DAX

```
Max Sales Target =  
CALCULATE(  
    MAX('Sales'[Target Sales]),  
    ALL('Sales')  
)
```

Final Thoughts:

This updated measure incorporates all your requested changes and should provide a dynamic, normalized bullet chart with consistent alignment and improved readability. The use of double underscores for variable names enhances clarity, and setting the font to Arial ensures consistency with your report's styling.


Please let me know if you need further adjustments or have any questions!

Thanks AI ODB - works perfectly!

You're welcome! I'm delighted to hear that the measure is working perfectly for you. If you have any more questions or need further assistance, don't hesitate to ask. Happy analyzing!

Wu-tang forever!

You're welcome! Wu-Tang forever!

From ChatGPT, designed with  FancyGPT