Custom Power BI Visual – Rounded Bar Chart

Overview

This project implements a **custom Power BI visual** using the **Power BI Visuals SDK**. The visual focuses on **business-friendly performance comparison** between **Actual Revenue and Target Revenue**, with strong emphasis on **visual clarity, conditional formatting, and interactivity**.

The visual is designed to be reusable across different categorical dimensions (e.g., Salesperson, Region) and supports Power BI slicers and cross-filtering.

---

1. How to Run Locally

Prerequisites

- **Node.js** (v18+ recommended)
- **npm**
- **Power BI Desktop**
- **Power BI Visuals SDK**

Verify installation:

```
node -v
npm -v
pbiviz --version
```

If `pbiviz` is not installed:

```
npm install -g powerbi-visuals-tools
```

Steps to Run the Visual Locally

Clone or unzip the source code folder:

```
roundedBarVisual/
```

1.

Navigate to the project root:

```
cd roundedBarVisual
```

2.

Install dependencies:

```
npm install
```

3.

Start the local development server:

```
pbiviz start
```

4.
5. Open **Power BI Desktop**

Enable developer mode:

```
File → Options → Security → Enable Developer Mode
```

6.
7. In the **Visuals pane**, select **Developer Visual**

8. The visual will connect to the local server and render live updates

---

2. How to Package the Visual (.pbiviz)

To generate the distributable package:

1. Stop the development server (Ctrl + C)

Run:

```
pbiviz package
```

2.

The packaged file will be generated at:

```
dist/RoundedBarVisual.pbiviz
```

3.

This `.pbiviz` file can be imported into any Power BI report using:

```
Visuals pane → Import a visual from a file
```

---

3. Features Implemented

3.1 Rounded Columns

- Columns have **rounded corners only on the relevant edge**:
    - Positive values → rounded **top** corners
    - Negative values → rounded **bottom** corners
- Radius is configurable from the formatting pane:
    - Range: **0–20 px**

3.2 Actual vs Target Comparison

- Bars represent **Actual Revenue**
- **Target reference markers** are drawn per category
- Target markers remain visible even when Actual exceeds Target

---

3.3 Conditional Formatting (Two Modes)

Mode A: Rule-Based

Based on **Achievement % = Actual / Target**

- `< Threshold 1` → Red
- `Threshold 1 to < Threshold 2` → Amber
- `≥ Threshold 2` → Green

Default thresholds:

- Threshold 1 = **0.80**
- Threshold 2 = **1.00**

Edge cases:

- Target is blank or 0 → treated as **N/A**
- Neutral color applied

Mode B: Field-Based

- If a valid **ColorHex** field is provided, the visual uses it directly
- Overrides rule-based coloring

Additional option:

- **Apply same color to data labels** (On/Off)

3.4 Interactive Tooltips

Tooltips display the following metrics on hover:

- Actual Revenue
- Target Revenue
- Achievement %
- Variance (absolute)
- Variance %

Tooltips are business-readable and update dynamically with filters and slicers.

3.5 Axes, Labels, and Layout

- Category labels displayed on X-axis
- Labels are **tilted** to avoid overlap for long names
- Value labels are displayed **above bars**
- Layout includes proper margins to avoid clipping

3.6 Cross-Filtering & Slicers

- Visual responds to:
    - Page-level slicers
    - Report-level filters
- Fully compatible with Power BI cross-filtering behavior

---

4. Limitations

- Sorting by derived metrics (e.g., Achievement %) must be handled via **DAX measures** in the Power BI model (recommended Power BI design).
- The visual assumes one category field at a time (e.g., Salesperson *or* Region).
- Extremely large category counts (>50) may reduce readability due to label density.

---

5. DAX Measures Used (PBIX Demo)

The following DAX measures were created in Power BI to support sorting and tooltip explanations:

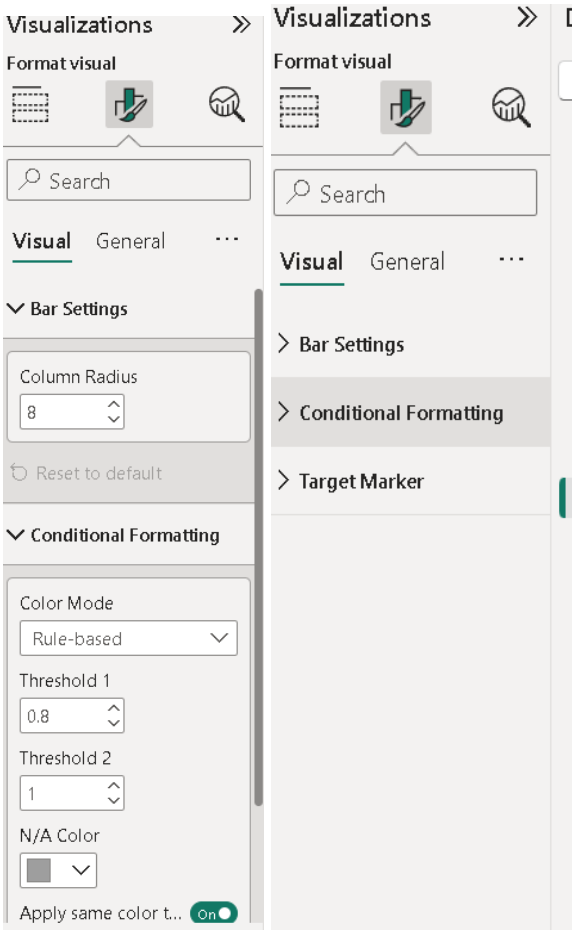
```
Achievement % =
DIVIDE(
    SUM(Sheet1[ActualRevenue]),
    SUM(Sheet1[TargetRevenue])
)

Variance =
SUM(Sheet1[ActualRevenue]) -
SUM(Sheet1[TargetRevenue])

Variance % =
DIVIDE(
    SUM(Sheet1[ActualRevenue]) - SUM(Sheet1[TargetRevenue]),
    SUM(Sheet1[TargetRevenue])
)
```

---

6. Formatting Pane Screenshot



This screenshot demonstrates formatting options including:

- Column radius
- Conditional formatting thresholds
- Target marker styling
- Data label color control

---

7. Technology Stack

- Power BI Visuals SDK
- TypeScript
- SVG rendering
- Power BI Formatting Model API

---

8. Conclusion

This custom visual demonstrates:

- Strong alignment with Power BI best practices
- Business-focused design
- Clean separation of semantic logic (DAX) and rendering logic (visual code)
- Extensibility for real-world enterprise reporting