

Waffle Chart - Deneb #3

Here is a detailed summary of the tables, columns, measures, and relationships in the provided Power BI model:

Tables and Columns:
Dates:

Columns: Date, Year, QuarterOfYear, MonthOfYear, DayOfMonth, DateInt, Month Name, Month & Year, Quarter & Year, DayOfWeek, DayOfWeekName, WeekEnding, MonthEnding, ISO Weeknumber, ISO Year, ISO Day of Year, Week & Year, WeeknYear, MonthnYear, QuarternYear, Fiscal Year, Fiscal Quarter, Fiscal Period, IsAfterToday, IsWorkingDay, IsHoliday, IsBusinessDay, Day Type, WeekOffset, MonthOffset, QuarterOffset, YearOffset, FiscalYearOffset, WeekCompleted, MonthCompleted, QuarterCompleted, WeekOfMonth, MonthRow, MonthColumn, MonthName, DayAbbrev, DayOfWeekNum_SunStart, MonthShort Sales:

Columns: OrderNumber, OrderDate, Customer Name Index, Channel, Currency Code, Warehouse Code, Delivery Region Index, Product Description Index, Order Quantity, Unit Price, Line Total, Total Unit Cost
Products:

Columns: Index, Product Name
Customers:

Columns: Customer Index, Customer Names, Emails, Head Office Index \u200bMeasures:

Measures: Total Sales, Total Quantity Ordered, SelectedYear, Number of Sales, Total Sales by Country, Sales Current YTD, Current Year Sales Target, Total Days Per Month, Current Year Total Days, Pct Target Sel MTD
ProductCombinations:

Columns: OrderNumber, Year, Combination, HasP1, HasP2, HasP3
CalendarHeatmapData:

Columns: Date, Year, Month, MonthName, DayOfMonth, DayOfWeekNum_SunStart, WeekOfMonth, IsCurrentMonth
Sales by Country:

Columns: Country, Sales Amount
Sales Targets:

Columns: Year, Target
WaffleCYSales:

Columns: Date, MonthOffset, YearOffset, MonthShort, TotSales, MonthOfYear,

Targ, IsAfterToday

Measures:

Total Sales: Represents total sales value.

Total Quantity Ordered: The total quantity ordered across all transactions.

SelectedYear: The year selected for analysis.

Number of Sales: The total number of sales transactions.

Total Sales by Country: Total sales aggregated by country.

Sales Current YTD: Year-to-date sales for the current year.

Current Year Sales Target: The sales target for the current year.

Total Days Per Month: Number of days in each month.

Current Year Total Days: Total number of days in the current year.

Pct Target Sel MTD: Percentage of the sales target achieved month-to-date, based on the selected month.

Relationships:

Sales[Product Description Index] to Products[Index]

Sales[Customer Name Index] to Customers[Customer Index]

Sales[OrderDate] to CalendarHeatmapData[Date]

Sales[OrderDate] to Dates[Date]

This summary should provide a solid basis for prompting an AI to generate scripts or insights based on the report's structure and data model.

Prompt for AI to Write Deneb Script:

Write a Deneb Vega-Lite JSON script to create a dynamic waffle chart that shows the percentage of the annual sales target achieved through the selected month. The waffle chart should represent this percentage by filling blue circles in a 10x10 grid.

Key Elements:

Input Data:

Measure Pct Target Sel MTD (DAX calculation) to determine the percentage of target sales achieved.

The user can select a month, and the chart should update dynamically to reflect the sales target achieved up to that month.

Structure of the Chart:

A 10x10 grid of circles should represent 100% achievement.

Each circle in the grid should represent 1% of the sales target.

The number of filled circles (in dark blue) should reflect the percentage calculated by the Pct Target Sel MTD measure.

Unfilled circles should remain in a light blue color.

The displayed percentage (numeric) should be dynamically updated based on the measure.

Additional Features:

Add a title dynamically showing "Percentage of Annual Sales Target Achieved Through [Selected Month]".

The selected month should be highlighted or labeled in the chart. Ensure the chart updates automatically based on the measure and selected month.

Certainly! Below is a Deneb Vega-Lite JSON script that creates a dynamic waffle chart to display the percentage of the annual sales target achieved through the selected month. This script assumes you have measures for `Pct Target Sel MTD` and `Selected Month` in your Power BI model.

Deneb Vega-Lite JSON Script:

```
json

{
  "$schema": "https://vega.github.io/schema/vega-lite/v5.json",
  "description": "Dynamic Waffle Chart for Sales Target Achievement",
  "data": {
    "name": "waffleData"
  },
  "transform": [
    {
      "window": [{"op": "row_number", "as": "id"}]
    },
    {
      "calculate": "(datum.id - 1) % 10",
      "as": "col"
    },
    {
      "calculate": "9 - floor((datum.id - 1) / 10)",
      "as": "row"
    },
    {
      "calculate": "datum.id <= round(datum.PctTargetSelMTD * 100) ? 'filled' : 'empty'",
      "as": "status"
    }
  ],
  "mark": {
    "type": "circle",
    "size": 200
  },
  "encoding": {
    "x": {
      "field": "col",
      "type": "ordinal",
      "axis": null
    },
    "y": {
      "field": "row",
      "type": "ordinal",
      "axis": null
    },
    "fill": {
      "field": "status",
      "type": "nominal",
      "scale": {
        "domain": ["filled", "empty"],
        "range": ["darkblue", "lightblue"]
      },
      "legend": null
    }
  },
  "title": {
    "text": [
      "Percentage of Annual Sales Target Achieved Through " + datum.SelectedMonth,
      round(datum.PctTargetSelMTD * 100) + "%"
    ],
    "fontSize": 16,
    "fontWeight": "bold",
    "subtitleFontSize": 24,
    "subtitleFontWeight": "bold"
  },
  "config": {
    "view": {
      "stroke": null
    }
  }
}
```

Instructions to Implement the Script in Deneb:

- 1 Create a New Deneb Visual:
 - 2 In Power BI, add a new Deneb visual to your report.
- 3 Set Up Data Fields:
 - 4 Measure Fields:
 - 5 Drag the `Pct Target Sel MTD` measure into the Deneb visual's **Fields** pane and rename it to `PctTargetSelMTD`.
 - 6 Drag the measure or field representing the selected month into the **Fields** pane and rename it to `SelectedMonth`.
 - 7 Create a Dummy Table:
 - 8 Since we need 100 data points for the 10x10 grid, create a new table in Power BI with at least 100 rows. This can be a simple table with a single column of numbers from 1 to 100.
 - 9 Add this table to the Deneb visual's **Fields** pane and rename it to `waffleData`.
 - 10 Adjust the Data Name in the Script:
 - 11 Ensure the `"data": { "name": "waffleData" }` in the script matches the name of the dataset in the Deneb visual.
 - 12 Bind Data Fields in Deneb:
 - 13 In Deneb's **Data Roles** settings, map the fields as follows:
 - 14 `waffleData`: Map to the dummy table with 100 rows.
 - 15 `PctTargetSelMTD`: Map to the `Pct Target Sel MTD` measure.
 - 16 `SelectedMonth`: Map to the selected month measure or field.
 - 17 Update Field References in the Script:
 - 18 Replace `datum.PctTargetSelMTD` and `datum.SelectedMonth` in the script with the exact field names if they differ.
 - 19 Test the Visual:
 - 20 The waffle chart should now display, filling dark blue circles up to the percentage specified by `Pct Target Sel MTD`.
 - 21 The title should dynamically show the selected month and the percentage achieved.

Additional Notes:

- Dynamic Title:
 - The `title` section uses concatenation to include dynamic values. Ensure that the fields `datum.SelectedMonth` and `datum.PctTargetSelMTD` are accessible in this context.
- Circle Grid Layout:
 - The calculation for `col` and `row` positions the circles to create a 10x10 grid, filling from left to right and bottom to top.
- Color Encoding:
 - The `fill` encoding uses the `status` field to determine if a circle should be `filled` (dark blue) or `empty` (light blue).
- Interactivity:
 - The chart updates automatically when the `Pct Target Sel MTD` measure or the selected month changes.

Troubleshooting:

- Field Accessibility:
 - If you encounter issues with fields not being recognized in the script, double-check the field names and ensure they are correctly mapped in Deneb's **Data Roles**.
- Data Volume:
 - The dummy table must contain at least 100 rows to generate the full grid.

- Calculation Adjustments:

- If the circles are not filling correctly, verify the calculation in the `transform` section, especially the rounding of the percentage.

Example DAX for Measures (If Needed):

- Pct Target Sel MTD:

DAX

```
Pct Target Sel MTD =  
DIVIDE(  
    [Sales Current YTD],  
    [Current Year Sales Target],  
    0  
)
```

- SelectedMonth:

DAX

```
SelectedMonth =  
SELECTEDVALUE('Dates'[Month Name], "All Months")
```

- Ensure that your date table '`Dates`' has a `[Month Name]` column.

Final Thoughts:

This script provides a dynamic and visually appealing way to track sales target achievements over the year. By integrating it with Power BI's slicers and filters, users can interactively select different months and see real-time updates in the waffle chart.

If you need further customization or run into any issues, feel free to ask for additional assistance!

```
{  
  "$schema": "https://vega.github.io/schema/vega-lite/v5.json",  
  "description": "Dynamic Waffle Chart for Sales Target Achievement",  
  "data": {  
    "name": "waffleData"  
  },  
  "transform": [  
    {  
      "window": [{"op": "row_number", "as": "id"}]  
    },  
    {  
      "calculate": "(datum.id - 1) % 10",  
      "as": "col"  
    },  
    {  
      "calculate": "9 - floor((datum.id - 1) / 10)",  
      "as": "row"  
    },  
    {  
      "calculate": "datum.id <= round(datum.PctTargetSelMTD * 100) ? 'filled' : 'empty'",  
      "as": "status"  
    }  
  ],  
  "mark": {  
    "type": "circle",  
    "size": 200
```

```

        },
        "encoding": {
            "x": {
                "field": "col",
                "type": "ordinal",
                "axis": null
            },
            "y": {
                "field": "row",
                "type": "ordinal",
                "axis": null
            },
            "fill": {
                "field": "status",
                "type": "nominal",
                "scale": {
                    "domain": ["filled", "empty"],
                    "range": ["darkblue", "lightblue"]
                },
                "legend": null
            },
            "title": {
                "text": [
                    "Percentage of Annual Sales Target Achieved Through " + datum.SelectedMonth,
                    round(datum.PctTargetSelMTD * 100) + "%"
                ],
                "fontSize": 16,
                "fontWeight": "bold",
                "subtitleFontSize": 24,
                "subtitleFontWeight": "bold"
            },
            "config": [
                "view": {
                    "stroke": null
                }
            ]
        }
    }
}

```

Problem with reeoudiung on line 20 and line 51

I'm sorry to hear you're experiencing issues with the rounding functions on line 20 and line 51 of your Deneb Vega-Lite JSON script. Let's work together to identify and resolve these problems so your dynamic waffle chart functions correctly.

Issue on Line 20:

Line 20:

```

        json
        "calculate": "datum.id <= round(datum.PctTargetSelMTD * 100) ? 'filled' : 'empty'",
```

Problem Explanation:

The error on line 20 is likely due to the `round` function not being recognized or properly used within the Vega expression language in the `calculate` transform. The Vega expression language supports mathematical functions, but sometimes functions like `round` need to be used carefully.

Solution:

Replace the `round` function with an alternative that ensures proper rounding. One effective method is to add `0.5` before applying the `floor` function, which simulates rounding to the nearest whole number.

Revised Line 20:

```
json
"calculate": "datum.id <= floor(datum.PctTargetSelMTD * 100 + 0.5) ? 'filled' : 'empty'",
```

This modification ensures that the percentage calculation rounds correctly by adding `0.5` before flooring, effectively simulating the `round` function.

Issue on Line 51:

Line 51 and 52:

```
json
"text": [
    "Percentage of Annual Sales Target Achieved Through " + datum.SelectedMonth,
    round(datum.PctTargetSelMTD * 100) + "%"
],
```

Problem Explanation:

The issue here is that `datum` is not accessible within the `title` property in Vega-Lite specifications. The `title` field does not support data expressions in the way marks and transforms do. Directly referencing `datum.SelectedMonth` and `datum.PctTargetSelMTD` in the `title` will not work because the `title` does not have access to the data scope.

Solution:

To use dynamic data in the title, you need to utilize `signals` or create a `text mark` that can access the data. Here's how you can adjust your script to incorporate signals for `SelectedMonth` and `PctTargetSelMTD`.

Revised Script with Solutions Applied:

```
json
{
    "$schema": "https://vega.github.io/schema/vega-lite/v5.json",
    "description": "Dynamic Waffle Chart for Sales Target Achievement",
    "signals": [
        {
            "name": "PctTargetSelMTD",
            "update": "PctTargetSelMTDData[0].PctTargetSelMTD"
        },
        {
            "name": "SelectedMonth",
            "update": "SelectedMonthData[0].SelectedMonth"
        }
    ],
    "data": {
        "name": "waffleData"
    },
    "transform": [
        {
            "window": [{"op": "row_number", "as": "id"}]
        },
        {
            "calculate": "(datum.id - 1) % 10",
            "as": "col"
        },
        {
            "calculate": "9 - floor((datum.id - 1) / 10)",
            "as": "row"
        }
    ]
}
```

```

        "calculate": "datum.id <= floor(PctTargetSelMTD * 100 + 0.5) ? 'filled' : 'empty'",  

        "as": "status"
    }  

],  

"mark": {  

    "type": "circle",  

    "size": 200
},  

"encoding": {  

    "x": {  

        "field": "col",  

        "type": "ordinal",  

        "axis": null
    },  

    "y": {  

        "field": "row",  

        "type": "ordinal",  

        "axis": null
    },  

    "fill": {  

        "field": "status",  

        "type": "nominal",  

        "scale": {  

            "domain": ["filled", "empty"],  

            "range": ["darkblue", "lightblue"]
        },
        "legend": null
    }
},  

"title": {  

    "text": "Percentage of Annual Sales Target Achieved Through " + SelectedMonth,  

    "fontSize": 16,  

    "fontWeight": "bold",
    "subtitle": {
        "text": (PctTargetSelMTD * 100).toFixed(0) + "%",
        "fontSize": 24,
        "fontWeight": "bold"
    }
},  

"config": {  

    "view": {  

        "stroke": null
    }
}
}

```

Implementation Steps in Deneb:

1. Add Measures to the Fields Pane:

- **Pct Target Sel MTD:**
 - Drag this measure into the **Fields** pane of the Deneb visual.
 - Deneb will create a dataset for it, accessible via `PctTargetSelMTDData`.
- **SelectedMonth:**
 - Drag the field or measure representing the selected month into the **Fields** pane.
 - Deneb will create a dataset for it, accessible via `SelectedMonthData`.

2. Adjust Signal Definitions:

- Ensure that the `update` properties in your `signals` match the dataset names created by Deneb.

json

```

"signals": [  

    {  

        "name": "PctTargetSelMTD",  

        "update": "PctTargetSelMTDData[0].PctTargetSelMTD"
    },  

    {  

        "name": "SelectedMonth",  

        "update": "SelectedMonthData[0].SelectedMonth"
    }
]

```

```
    }  
],
```

3. Modify the `calculate` Expressions:

- Use the signals `PctTargetSelMTD` and `SelectedMonth` in your `calculate` and `title` expressions.

json

```
{  
  "calculate": "datum.id <= floor(PctTargetSelMTD * 100 + 0.5) ? 'filled' : 'empty'",  
  "as": "status"  
}
```

4. Update the `title` Property:

- Modify the `title` to use the signals.

json

```
"title": {  
  "text": "Percentage of Annual Sales Target Achieved Through " + SelectedMonth,  
  "fontSize": 16,  
  "fontWeight": "bold",  
  "subtitle": {  
    "text": (PctTargetSelMTD * 100).toFixed(0) + "%",  
    "fontSize": 24,  
    "fontWeight": "bold"  
  }  
}
```

- The `toFixed(0)` function ensures that the percentage is rounded to the nearest whole number.

5. Verify Field Names and Data Binding:

- Ensure that field names in your script exactly match those in Deneb, including case sensitivity.
- Confirm that the measures are correctly bound to the datasets or signals.

6. Test the Visual:

- The waffle chart should now display correctly, with the percentage rounded appropriately.
- The title and subtitle should dynamically update based on the selected month and the `Pct Target Sel MTD` measure.

Additional Notes:

- Function Availability:**
 - The Vega expression language supports `floor`, `toFixed`, and basic arithmetic operations.
 - Ensure you're using functions compatible with the Vega expression language.
- Data Volume:**
 - Your `waffleData` dataset should have at least 100 rows to represent a 10x10 grid.
 - You can create a calculated table in Power BI with numbers from 1 to 100 for this purpose.
- Interactivity:**
 - The chart will update automatically when the `Pct Target Sel MTD` measure or the selected month changes in your data model.

Troubleshooting Tips:

- If the Circles Aren't Filling Correctly:**
 - Double-check the calculation in the `transform` section.
 - Ensure that `PctTargetSelMTD` returns a decimal between 0 and 1.
- If the Title Doesn't Update:**
 - Verify that the signals are correctly defined and that their `update` properties point to the correct datasets.

- Ensure that the measures are properly added to the Deneb visual's Fields pane.
- If You Receive Errors About Functions:
 - Confirm that you're using functions supported by the Vega expression language.
 - Replace unsupported functions with equivalent expressions.

Example DAX Measures (If Needed):

Pct Target Sel MTD:

DAX

```
Pct Target Sel MTD =
DIVIDE(
    [Sales Current YTD],
    [Current Year Sales Target],
    0
)
```

SelectedMonth:

DAX

```
SelectedMonth =
SELECTEDVALUE('Dates'[Month Name], "All Months")
```

- Ensure that your 'Dates' table has a [Month Name] column.

Final Thoughts:

By adjusting the rounding method on line 20 and utilizing signals for dynamic data in the title on line 51, your Deneb Vega-Lite script should function as intended. This approach ensures that the percentage is accurately rounded and that dynamic titles reflect the selected data.

If you have further questions or encounter additional issues, please don't hesitate to ask for more assistance. I'm here to help you get your dynamic waffle chart working perfectly!

Problem line 59

Trailing comma(519)

I apologize for the issues you're encountering with the script, particularly on line 59 where you're receiving a "Trailing comma" error with [json\(519\)](#). This error indicates there's a syntax issue in the JSON code, specifically an extra comma or invalid syntax that's causing the JSON parser to fail.

Let's address the problem step by step to resolve the error and ensure your Deneb Vega-Lite JSON script works correctly.

Identifying the Problem on Line 59

Line 59:

json

```
"title": {
```

The error message suggests there's a trailing comma or syntax error near this line. Upon reviewing the script, the issue arises from how expressions are being used within the `title` and `subtitle` properties. JSON requires strict

syntax rules, and expressions need to be properly formatted to avoid parsing errors.

Explanation of the Issue

1. Invalid Use of Expressions in JSON

In the JSON format, values must be of type string, number, object, array, `true`, `false`, or `null`. You cannot directly include JavaScript expressions like `(PctTargetSelMTD * 100).toFixed(0) + "%".` Such expressions need to be enclosed within a string and appropriately handled by Vega-Lite using signals or expression syntax.

2. Trailing Comma

A trailing comma in JSON refers to an extra comma after the last item in an object or array. While some languages tolerate trailing commas, JSON does not. However, in this case, the primary issue is the invalid expression syntax rather than a literal trailing comma.

Correcting the Script

1. Use of Signals and Expressions

To fix the error, we'll modify the `title` and `subtitle` properties to use Vega-Lite's expression capabilities properly. We'll wrap the expressions in strings and use the `"signal"` property to indicate that the string should be evaluated as an expression.

2. Ensuring Proper JSON Syntax

We'll ensure that all expressions are enclosed within double quotes and that any special characters are properly escaped. We'll also check for any actual trailing commas and remove them if present.

Revised Script

Here's the corrected script with the necessary adjustments:

```
json

{
  "$schema": "https://vega.github.io/schema/vega-lite/v5.json",
  "description": "Dynamic Waffle Chart for Sales Target Achievement",
  "signals": [
    {
      "name": "PctTargetSelMTD",
      "update": "PctTargetSelMTDData[0].PctTargetSelMTD"
    },
    {
      "name": "SelectedMonth",
      "update": "SelectedMonthData[0].SelectedMonth"
    }
  ],
  "data": {
    "name": "WaffleData"
  },
  "transform": [
    {
      "window": [{"op": "row_number", "as": "id"}]
    },
    {
      "calculate": "(datum.id - 1) % 10",
      "as": "col"
    },
    {
      "calculate": "9 - floor((datum.id - 1) / 10)",
      "as": "row"
    },
    {
      "calculate": "datum.id <= floor(PctTargetSelMTD * 100 + 0.5) ? 'filled' : 'empty'",
      "as": "status"
    }
  ],
  "mark": {
    "type": "circle",
    "size": 200
  },
  "encoding": {
    "x": {
      "field": "col",
      "type": "ordinal",
      "signal": "PctTargetSelMTD"
    },
    "y": {
      "field": "row",
      "type": "ordinal",
      "signal": "SelectedMonth"
    }
  }
}
```

```

        "axis": null
    },
    "y": {
        "field": "row",
        "type": "ordinal",
        "axis": null
    },
    "fill": {
        "field": "status",
        "type": "nominal",
        "scale": {
            "domain": ["filled", "empty"],
            "range": ["darkblue", "lightblue"]
        },
        "legend": null
    }
},
"title": {
    "text": {
        "signal": "\"Percentage of Annual Sales Target Achieved Through \" + SelectedMonth"
    },
    "fontSize": 16,
    "fontWeight": "bold",
    "subtitle": {
        "text": {
            "signal": "format(PctTargetSelMTD * 100, '.0f') + \"%\""
        },
        "fontSize": 24,
        "fontWeight": "bold"
    }
},
"config": {
    "view": {
        "stroke": null
    }
}
}

```

Explanation of Changes

1. Adjusted title.text and subtitle.text Properties

- title.text:

```

        json

    "text": {
        "signal": "\"Percentage of Annual Sales Target Achieved Through \" + SelectedMonth"
    },

```

- Wrapped the entire expression in a string using double quotes.
- Used the `signal` property to indicate that it's an expression to be evaluated.
- Escaped inner double quotes with backslashes if necessary.

- subtitle.text:

```

        json

    "text": {
        "signal": "format(PctTargetSelMTD * 100, '.0f') + \"%\""
    },

```

- Used the `format` function to format the percentage value to zero decimal places.
- Wrapped the percentage symbol `%` in double quotes.
- Ensured the entire expression is a valid string for the `signal`.

2. Ensured Proper JSON Syntax

- Checked all properties to make sure there are no trailing commas.
- Verified that all strings are enclosed in double quotes.

- Ensured that expressions within the JSON are properly formatted as strings and use Vega-Lite's expression syntax.

Implementation Steps in Deneb

1. Add Measures to the Fields Pane

- PctTargetSelMTD:
 - Drag this measure into the Deneb visual's **Fields** pane.
 - It will be accessible via `PctTargetSelMTDData`.
- SelectedMonth:
 - Drag this measure or field into the Deneb visual's **Fields** pane.
 - It will be accessible via `SelectedMonthData`.

2. Define Signals Correctly

- The `signals` array correctly references these measures:

json

```
"signals": [  
  {  
    "name": "PctTargetSelMTD",  
    "update": "PctTargetSelMTDData[0].PctTargetSelMTD"  
  },  
  {  
    "name": "SelectedMonth",  
    "update": "SelectedMonthData[0].SelectedMonth"  
  }  
,  
],
```

3. Modify the title and subtitle Properties

- `title.text` uses a `signal` to concatenate the static text with the `SelectedMonth` value.
- `subtitle.text` uses a `signal` to calculate and format the percentage.

4. Verify No Trailing Commas

- Ensured that there are no commas after the last property in an object. For example, after `"fontWeight": "bold"` in both the `title` and `subtitle`, there should be no comma unless more properties follow.

Testing the Visual

- After implementing the script in Deneb, the waffle chart should render without any JSON parsing errors.
- The `title` should display the correct text, dynamically showing the selected month.
- The `subtitle` should display the percentage of the sales target achieved, rounded to the nearest whole number.
- The chart should update dynamically based on user selections and data changes.

Troubleshooting Tips

- JSON Parsing Errors:**
 - If you still encounter JSON parsing errors, double-check the syntax around the areas where expressions are used.
 - Ensure that all strings are properly enclosed in double quotes and that any inner double quotes are escaped.
- Expression Errors:**
 - Make sure that the expressions within `signal` properties are valid and use functions supported by Vega-Lite.
 - Use `format()` instead of `toFixed()` for formatting numbers in Vega-Lite expressions.
- Data Binding Issues:**

- Confirm that the field names in the script match exactly (including case sensitivity) with those in Deneb's Fields pane.
- Verify that the datasets (`PctTargetSelMTDData`, `SelectedMonthData`, `waffleData`) are correctly referenced.

Final Thoughts

By correcting the syntax in the `title` and `subtitle` properties and ensuring proper JSON formatting, the script should now work without errors. The use of `signal` allows you to include dynamic expressions in your title and subtitle, and the `format()` function helps in displaying the percentage correctly.

If you have any more questions or need further assistance, please feel free to ask. I'm here to help you get your dynamic waffle chart working perfectly!

Additional Resources

- Vega-Lite Documentation on Text and Signals:
 - Text Mark Properties
 - Using Signals in Vega-Lite
- Formatting Numbers in Vega-Lite:
 - Use the `format()` function to format numbers within expressions.

Example DAX Measures (If Needed)

`PctTargetSelMTD:`

DAX

```
PctTargetSelMTD =
DIVIDE(
    [Sales Current YTD],
    [Current Year Sales Target],
    0
)
```

`SelectedMonth:`

DAX

```
SelectedMonth =
SELECTEDVALUE('Dates'[Month Name], "All Months")
```

I hope this helps resolve the error and get your visual up and running. Let me know if you need any further assistance!

Change title to ""Percentage of Annual Sales Target Achieved" and eliminate subtitle

Certainly! I'll update the Deneb Vega-Lite JSON script to change the title to "`Percentage of Annual Sales Target Achieved`" and eliminate the subtitle. This adjustment will make the title static and remove any dynamic elements related to the selected month or percentage in the subtitle.

Below is the revised script:

Updated Deneb Vega-Lite JSON Script

json

```
{
  "$schema": "https://vega.github.io/schema/vega-lite/v5.json",
  "description": "Dynamic Waffle Chart for Sales Target Achievement",
  "signals": [
    {
      "name": "PctTargetSelMTD",
      "update": "PctTargetSelMTDData[0].PctTargetSelMTD"
    }
  ],
  "data": {
    "name": "waffleData"
  },
  "transform": [
    {
      "window": [{"op": "row_number", "as": "id"}]
    },
    {
      "calculate": "(datum.id - 1) % 10",
      "as": "col"
    },
    {
      "calculate": "9 - floor((datum.id - 1) / 10)",
      "as": "row"
    },
    {
      "calculate": "datum.id <= floor(PctTargetSelMTD * 100 + 0.5) ? 'filled' : 'empty'",
      "as": "status"
    }
  ],
  "mark": {
    "type": "circle",
    "size": 200
  },
  "encoding": {
    "x": {
      "field": "col",
      "type": "ordinal",
      "axis": null
    },
    "y": {
      "field": "row",
      "type": "ordinal",
      "axis": null
    },
    "fill": {
      "field": "status",
      "type": "nominal",
      "scale": {
        "domain": ["filled", "empty"],
        "range": ["darkblue", "lightblue"]
      },
      "legend": null
    }
  },
  "title": {
    "text": "Percentage of Annual Sales Target Achieved",
    "fontSize": 16,
    "fontWeight": "bold"
  },
  "config": {
    "view": {
      "stroke": null
    }
  }
}
```

Explanation of Changes

1. Updated the Title

- Removed the Dynamic Elements:
 - Changed the `title.text` property from a dynamic expression to a static string.
- New Title Definition:

```
json

  "title": {
    "text": "Percentage of Annual Sales Target Achieved",
    "fontSize": 16,
    "fontWeight": "bold"
  },
```

2. Eliminated the Subtitle

- Removed the `subtitle` Property:
 - The `subtitle` property has been removed entirely, eliminating any subtitle from the chart.

3. Adjusted Signals

- Removed Unnecessary Signals:
 - Since we're no longer using `SelectedMonth` in the title or subtitle, we can remove the `SelectedMonth` signal.
- Updated Signals Section:

```
json

  "signals": [
    {
      "name": "PctTargetSelMTD",
      "update": "PctTargetSelMTDData[0].PctTargetSelMTD"
    }
  ],
```

4. Ensured Proper JSON Syntax

- Verified Commas and Syntax:
 - Ensured there are no trailing commas or syntax errors in the updated script.

Implementation Steps in Deneb

1. Add Measures to the Fields Pane

- `PctTargetSelMTD`:
 - Drag this measure into the Deneb visual's **Fields** pane.
 - It will be accessible via `PctTargetSelMTDData`.
- Removed `SelectedMonth`:
 - Since we're no longer using `SelectedMonth`, you don't need to include this measure in the Fields pane.

2. Verify the `waffleData` Dataset

- Ensure that the `waffleData` dataset contains at least 100 rows to represent the 10x10 grid.

3. Update the Signals

- The `signals` section now only includes `PctTargetSelMTD`:

```
json

  "signals": [
    {
      "name": "PctTargetSelMTD",
      "update": "PctTargetSelMTDData[0].PctTargetSelMTD"
    }
  ],
```

4. Check the Transform Calculations

- The calculations in the `transform` section remain the same, using `PctTargetSelMTD` to determine how many circles to fill.

5. Test the Visual

- After implementing the script, the waffle chart should display correctly.
- The title should now be static, showing "Percentage of Annual Sales Target Achieved".

- The chart should still dynamically fill the circles based on the `PctTargetSelMTD` measure.

Additional Notes

- Interactivity Remains:**
 - Even though the title is now static, the waffle chart will continue to update dynamically based on changes to the `PctTargetSelMTD` measure.
- No Dependency on Selected Month:**
 - By removing the dynamic elements related to the selected month, the chart is simplified and focuses solely on the percentage of the annual sales target achieved.
- Simplified Script:**
 - The script is now more straightforward, reducing potential errors related to dynamic expressions in the title and subtitle.

Troubleshooting Tips

- If the Circles Aren't Filling Correctly:**
 - Verify that the `PctTargetSelMTD` measure returns a decimal value between 0 and 1.
 - Ensure that the measure is correctly bound in Deneb's Fields pane and accessible via `PctTargetSelMTDData`.
- If the Title Doesn't Display:**
 - Check that the `title` property is correctly defined and that there are no syntax errors.
- Data Binding Issues:**
 - Confirm that all field names in the script match exactly with those in Deneb's Fields pane.

Example DAX Measure for `PctTargetSelMTD` (If Needed)

DAX

```
PctTargetSelMTD =
DIVIDE(
    [Sales Current YTD],
    [Current Year Sales Target],
    0
)
```

- Ensure that this measure calculates the percentage of the annual sales target achieved to date.

Final Thoughts

With these changes, your Deneb Vega-Lite JSON script should now display a waffle chart with a static title and no subtitle, as per your request. The visual will continue to dynamically represent the percentage of the annual sales target achieved by filling the appropriate number of circles in the grid.

If you have any further questions or need additional assistance, please don't hesitate to ask!

line 3 Property signals is not allowed.

I understand that you're encountering the error: "line 3 Property signals is not allowed." This error occurs because, in Vega-Lite (which Deneb uses), the `signals` property is not allowed at the top level of the specification. Unlike Vega,

Vega-Lite doesn't support top-level signals; instead, it uses **parameters** and data transformations to achieve similar functionality.

Let's work together to adjust your script to remove the **signals** property and find an alternative method to incorporate your **PctTargetSelMTD** measure into the calculations. We'll ensure the script is valid Vega-Lite JSON and that it functions correctly within Deneb.

Understanding the Issue

- **Vega vs. Vega-Lite:** In Vega, you can use **signals** at the top level to define dynamic variables. However, Vega-Lite does not support top-level **signals**. Instead, Vega-Lite uses **params** (parameters) and other mechanisms for interactivity and dynamic calculations.
- **Deneb and Measures:** In Deneb, when you add measures to the **Fields** pane, they are accessible within your Vega-Lite specification as data fields. We can leverage this to include your **PctTargetSelMTD** measure directly in the data transformations.

Solution Overview

We will:

- 1 Remove the **signals** property from the script.
- 2 Modify the data transformations to access **PctTargetSelMTD** directly from the data.
- 3 Ensure the measure is accessible within the data scope by adding it appropriately in Deneb.
- 4 Adjust the script accordingly to use the measure without **signals**.

Step-by-Step Solution

1. Remove the **signals** Property

Delete the entire **signals** section from your script:

```
json  
  
"signals": [  
  {  
    "name": "PctTargetSelMTD",  
    "update": "PctTargetSelMTDData[0].PctTargetSelMTD"  
  }  
,
```

2. Modify the Data Transformations

Since we can't use **signals**, we'll adjust the **transform** section to access **PctTargetSelMTD** directly from the data. We'll ensure that **PctTargetSelMTD** is available as a field in the data.

Here's how to adjust the **transform**:

- Use a **cross** or **joinaggregate** **transform** to merge the measure into the data scope.
- Alternatively, since **PctTargetSelMTD** is a single value, we can use a **lookup** **transform** or include it as a constant.

3. Ensure **PctTargetSelMTD** is Accessible in the Data

In Deneb, when you add measures to the **Fields** pane, they are accessible as data sources. We'll make sure that **PctTargetSelMTD** is added and accessible.

Implementation in Deneb:

- Add **PctTargetSelMTD** to the **Fields** Pane:
 - Drag the **PctTargetSelMTD** measure into the Deneb visual's **Fields** pane.
 - Deneb will create a dataset, let's call it **PctTargetSelMTDData**.
- Create a Calculated Table for the Waffle Chart:
 - Create a calculated table in Power BI with 100 rows (numbers from 1 to 100).
 - For simplicity, let's name it **WaffleData**.
 - Add a column called **id** with values from 1 to 100.
- Add **WaffleData** to the Deneb Visual:

- Add `waffleData` to the Fields pane in Deneb.

4. Adjust the Vega-Lite Specification

Now, modify the script to access `PctTargetSelMTD` from the data.

Revised Script:

```
json

{
  "$schema": "https://vega.github.io/schema/vega-lite/v5.json",
  "description": "Dynamic Waffle Chart for Sales Target Achievement",
  "data": {
    "name": "waffleData"
  },
  "transform": [
    {
      "calculate": "(datum.id - 1) % 10",
      "as": "col"
    },
    {
      "calculate": "9 - floor((datum.id - 1) / 10)",
      "as": "row"
    },
    {
      "joinaggregate": [
        {
          "op": "mean",
          "field": "PctTargetSelMTD",
          "as": "PctTargetSelMTDValue"
        }
      ]
    },
    {
      "calculate": "datum.id <= floor(datum.PctTargetSelMTDValue * 100 + 0.5) ? 'filled' : 'empty'",
      "as": "status"
    }
  ],
  "mark": {
    "type": "circle",
    "size": 200
  },
  "encoding": {
    "x": {
      "field": "col",
      "type": "ordinal",
      "axis": null
    },
    "y": {
      "field": "row",
      "type": "ordinal",
      "axis": null
    },
    "fill": {
      "field": "status",
      "type": "nominal",
      "scale": {
        "domain": ["filled", "empty"],
        "range": ["darkblue", "lightblue"]
      },
      "legend": null
    },
    "title": {
      "text": "Percentage of Annual Sales Target Achieved",
      "fontSize": 16,
      "fontWeight": "bold"
    },
    "config": {
      "view": {
        "stroke": null
      }
    }
  }
}
```

Explanation of Changes

1. Data Source

- Data Name: "name": "waffleData" references the `WaffleData` table you added to Deneb.

2. Transform Adjustments

- Removed `window` Transform:
 - Since your `WaffleData` already has an `id` field from 1 to 100, you don't need to generate it.
- Calculations for `col` and `row`:
 - These remain the same to position the circles in a 10x10 grid.
- `joinaggregate` Transform:
 - This transform adds the `PctTargetSelMTD` value to each row of `waffleData`.
 - It computes the mean (since it's a single value, mean is appropriate) of `PctTargetSelMTD` and adds it as `PctTargetSelMTDValue` to each data point.
- Calculation of `status`:
 - Adjusted to use `datum.PctTargetSelMTDValue` instead of the signal.
 - This way, each data point knows the `PctTargetSelMTD` value.

3. Removed `signals` Property

- No `signals`: The script no longer uses `signals`, adhering to Vega-Lite's requirements.

4. Measure Access

- Measure as Data Field:
 - `PctTargetSelMTD` is accessed as a data field in the `joinaggregate` transform.

Implementation Steps in Deneb

1. Prepare the Data

- Create `WaffleData` Table in Power BI:

DAX

```
WaffleData =  
ADDCOLUMNS(  
    GENERATESERIES(1, 100, 1),  
    "id", [Value]  
)
```

- Add `WaffleData` to Deneb:

- Drag the `id` field from `WaffleData` into the Deneb visual's **Fields** pane.

2. Add `PctTargetSelMTD` Measure to Deneb

- Drag the `PctTargetSelMTD` measure into the Deneb visual's **Fields** pane.
- Ensure it is accessible in the data.

3. Verify Field Names

- Ensure that the field names in the script match the field names in Deneb, including case sensitivity.

4. Bind Data in Deneb

- Data Mappings:
 - `waffleData`: Map to `WaffleData` with the `id` field.
 - `PctTargetSelMTD`: Ensure it's included and accessible as a field.

5. Test the Visual

- The waffle chart should display, filling dark blue circles up to the percentage specified by `PctTargetSelMTD`.
- The title should display "Percentage of Annual Sales Target Achieved."

Additional Notes

Using `joinaggregate`

- The `joinaggregate` transform allows us to compute an aggregate value (like mean) and add it to each data point. Since `PctTargetSelMTD` is a single value, computing its mean simply provides the same value.

Accessing Measures in Transforms

- By adding the measure to the Fields pane, Deneb makes it accessible in the data scope, allowing us to use it in transforms.

Avoiding signals

- Vega-Lite specifications do not support top-level `signals`. Using data transformations and parameters is the correct approach.

Troubleshooting Tips

- If `PctTargetSelMTD` Is Not Recognized:
 - Ensure that `PctTargetSelMTD` is properly added to the Deneb visual's Fields pane.
 - Confirm that it is accessible in the data transformations.
- Data Field Names:
 - Verify that all field names in the script (`id`, `PctTargetSelMTD`, etc.) match exactly with those in your data.
- Calculations Not Working:
 - Check the `calculate` expressions for syntax errors.
 - Ensure that `PctTargetSelMTD` returns a decimal between 0 and 1.
- Visual Not Rendering Correctly:
 - Ensure that your `WaffleData` table has exactly 100 rows with `id` values from 1 to 100.

Final Thoughts

By removing the `signals` property and adjusting the data transformations to include your measure, we've created a valid Vega-Lite specification that should work in Deneb. The key was to ensure that `PctTargetSelMTD` is accessible within the data scope, which we achieved using the `joinaggregate` transform.

Complete Revised Script

Here's the complete script again for clarity:

```
json

{
  "$schema": "https://vega.github.io/schema/vega-lite/v5.json",
  "description": "Dynamic Waffle Chart for Sales Target Achievement",
  "data": {
    "name": "waffleData"
  },
  "transform": [
    {
      "calculate": "(datum.id - 1) % 10",
      "as": "col"
    },
    {
      "calculate": "9 - floor((datum.id - 1) / 10)",
      "as": "row"
    },
    {
      "joinaggregate": [
        {
          "op": "mean",
          "field": "PctTargetSelMTD",
          "as": "PctTargetSelMTDValue"
        }
      ]
    },
    {
      "calculate": "datum.id <= floor(datum.PctTargetSelMTDValue * 100 + 0.5) ? 'filled' : 'empty'",
      "as": "status"
    }
  ]
}
```

```
],
"mark": {
  "type": "circle",
  "size": 200
},
"encoding": {
  "x": {
    "field": "col",
    "type": "ordinal",
    "axis": null
  },
  "y": {
    "field": "row",
    "type": "ordinal",
    "axis": null
  },
  "fill": {
    "field": "status",
    "type": "nominal",
    "scale": {
      "domain": ["filled", "empty"],
      "range": ["darkblue", "lightblue"]
    },
    "legend": null
  }
},
"title": {
  "text": "Percentage of Annual Sales Target Achieved",
  "fontSize": 16,
  "fontWeight": "bold"
},
"config": {
  "view": {
    "stroke": null
  }
}
}
```

Example DAX for WaffleData and PctTargetSelMTD

WaffleData Table:

DAX

```
WaffleData =
ADDCOLUMNS(
  GENERATESERIES(1, 100, 1),
  "id", [Value]
)
```

PctTargetSelMTD Measure:

DAX

```
PctTargetSelMTD =
DIVIDE(
  [Sales Current YTD],
  [Current Year Sales Target],
  0
)
```

Conclusion

By adjusting the script and utilizing data transformations within Vega-Lite, we've successfully removed the unsupported `signals` property and ensured that your measure is accessible within the data scope. This approach aligns with Vega-Lite's specifications and should resolve the error you're experiencing.

If you have any further questions or need additional assistance, please don't hesitate to ask!

I understand that you're encountering the error "line 4 Property signals is not allowed." when trying to use the script I provided. This error occurs because Vega-Lite, which Deneb uses, does not support the `signals` property at the top level of the specification. Instead, Vega-Lite uses `parameters` (`params`) and data transformations to achieve similar functionality.

Let's work together to adjust your script to remove the `signals` property and find an alternative method to incorporate your `PctTargetSelMTD` measure into the calculations. I'll provide a revised script and detailed steps to help you implement it successfully in Deneb.

Understanding the Issue

- **Vega-Lite Limitations:** Vega-Lite does not support the `signals` property at the top level. Using it results in the error you're seeing.
- **Deneb Data Binding:** In Deneb, when you add fields and measures to the `Fields` pane, they are included in the same data source (`dataset`) for the visual. This means you can access both your table fields and measures directly in the Vega-Lite specification.

Solution Overview

We will:

- 1 Remove the `signals` property from the script.
- 2 Access `PctTargetSelMTD` directly from the data, since it's included in the same dataset.
- 3 Adjust the script to use the combined dataset for both the waffle grid and the measure.
- 4 Ensure proper data transformations to calculate the grid positions and fill status.

Revised Deneb Vega-Lite JSON Script

```
json
{
  "$schema": "https://vega.github.io/schema/vega-lite/v5.json",
  "description": "Dynamic Waffle Chart for Sales Target Achievement",
  "data": {
    "name": "dataset"
  },
  "transform": [
    {
      "calculate": "(datum.id - 1) % 10",
      "as": "col"
    },
    {
      "calculate": "9 - floor((datum.id - 1) / 10)",
      "as": "row"
    },
    {
      "calculate": "datum.id <= floor(datum.PctTargetSelMTD * 100 + 0.5) ? 'filled' : 'empty'",
      "as": "status"
    }
  ],
  "mark": {
    "type": "circle",
    "size": 200
  },
  "encoding": {
    "x": {
      "field": "col",
      "type": "ordinal",
      "axis": null
    },
    "y": {
      "field": "row",
      "type": "ordinal",
      "axis": null
    }
  }
}
```

```

        "type": "ordinal",
        "axis": null
    },
    "fill": {
        "field": "status",
        "type": "nominal",
        "scale": {
            "domain": ["filled", "empty"],
            "range": ["darkblue", "lightblue"]
        },
        "legend": null
    }
},
"title": {
    "text": "Percentage of Annual Sales Target Achieved",
    "fontSize": 16,
    "fontWeight": "bold"
},
"config": {
    "view": {
        "stroke": null
    }
}
}
}

```

Explanation of Changes

1. Data Source

- **Data Name:** Changed the data source to `"name": "dataset"` to reflect the combined dataset that Deneb creates when you add multiple fields and measures to the Fields pane.
- **No `signals`:** Removed the `signals` property entirely.

2. Transform Adjustments

- **Calculations for `col` and `row`:** Remain the same to position the circles in a 10x10 grid.
- **Calculation of `status`:**
 - Directly use `datum.PctTargetSelMTD` to determine whether each circle should be filled or empty.
 - The expression `datum.id <= floor(datum.PctTargetSelMTD * 100 + 0.5)` compares the `id` to the rounded percentage value.

3. Measure Access

- **Accessing `PctTargetSelMTD`:**
 - Since both `id` and `PctTargetSelMTD` are in the same dataset, you can access them directly in the transforms and encodings.

Implementation Steps in Deneb

1. Prepare the Data

- Create `WaffleData` Table in Power BI:

DAX

```

WaffleData =
GENERATESERIES(1, 100, 1)

```

- This table has a single column `Value` with values from 1 to 100.
- Rename the `Value` column to `id` for consistency.
- Add a Calculated Column for `id`:

DAX

```

id = [Value]

```

- Note: Alternatively, you can directly name the column `id` in the `GENERATESERIES` function.

2. Add Fields to Deneb

- Add `id` from `WaffleData`:

- Drag the `id` field into the Deneb visual's **Fields** pane.
- Add `PctTargetSelMTD` Measure:
 - Drag the `PctTargetSelMTD` measure into the Deneb visual's **Fields** pane.
- Deneb's Combined Dataset:
 - Deneb will combine these fields into a single dataset named `dataset`.

3. Verify Field Names

- Ensure that the field names in the script (`id`, `PctTargetSelMTD`) match exactly with those in Deneb, including case sensitivity.

4. Adjust the Vega-Lite Specification

- Use the combined `dataset` as the data source.
- Access `datum.id` and `datum.PctTargetSelMTD` directly in the transforms.

5. Test the Visual

- The waffle chart should display, filling dark blue circles up to the percentage specified by `PctTargetSelMTD`.
- The title should display "Percentage of Annual Sales Target Achieved."

Additional Notes

Deneb Data Binding

- When you add multiple fields (columns and measures) to the Fields pane, Deneb creates a single dataset (`dataset`) that includes all of them.
- This allows you to access both the `id` and `PctTargetSelMTD` directly in your Vega-Lite specification.

No Need for `signals` or `params`

- Since the measure is included in the dataset, you don't need to use `signals` or `params` to access it.
- This approach aligns with Vega-Lite's specifications and avoids the error you're encountering.

Calculations

- Grid Positioning:
 - The `col` and `row` calculations position each circle in the 10x10 grid.
- Fill Status:
 - The `status` calculation determines whether each circle is 'filled' or 'empty' based on `PctTargetSelMTD`.

Troubleshooting Tips

- If `PctTargetSelMTD` Is Not Recognized:
 - Ensure that `PctTargetSelMTD` is properly added to the Deneb visual's Fields pane.
 - Confirm that the field names in the script match exactly with those in Deneb.
- Circles Aren't Filling Correctly:
 - Verify that `PctTargetSelMTD` returns a decimal value between 0 and 1.
 - Check the calculation in the `transform` section for syntax errors.
- Visual Not Rendering:
 - Ensure that your `WaffleData` table has exactly 100 rows with `id` values from 1 to 100.
 - Confirm there are no syntax errors in the script.

Example DAX for `WaffleData` and `PctTargetSelMTD`

`WaffleData` Table:

DAX

```
WaffleData = GENERATESERIES(1, 100, 1)
```

- Rename the `value` Column to `id`:

DAX

```
id = [Value]
```

PctTargetSelMTD Measure:

DAX

```
PctTargetSelMTD =  
DIVIDE(  
    [Sales Current YTD],  
    [Current Year Sales Target],  
    0  
)
```

- Ensure that this measure calculates the percentage of the annual sales target achieved to date and returns a decimal between 0 and 1.

Final Thoughts

By removing the `signals` property and leveraging Deneb's ability to include fields and measures in a single dataset, we've adjusted the script to avoid the error and function correctly. This approach adheres to Vega-Lite's specifications and allows you to create the dynamic waffle chart as intended.

Summary of Steps

- ① Remove `signals` Property:
 - ② Eliminated the unsupported `signals` property from the script.
- ③ Use Combined Dataset:
 - ④ Access both `id` and `PctTargetSelMTD` from the same `dataset`.
- ⑤ Adjust Transforms:
 - ⑥ Modify the `transform` calculations to use `datum.PctTargetSelMTD`.
- ⑦ Implement in Deneb:
 - ⑧ Add `id` and `PctTargetSelMTD` to the Fields pane.
 - ⑨ Use the script as provided.

Need Further Assistance?

If you have any more questions or need additional help implementing this solution, please don't hesitate to ask. I'm here to help you get your dynamic waffle chart working perfectly!

Ok, pretty close. Everything seems to be calculating well. Let's make the circles bigger by 50%, add a little more spacing between them.

Also the value of `PctTargetSelMTD` in the form of for example "43%" should show up centered over the top to rows of circles on the waffle chart

I'm glad to hear that the chart is calculating correctly! Based on your new requirements, I've updated the Deneb Vega-Lite JSON script to:

- ① Increase the circle size by 50% and add more spacing between them.

- 2 Display the value of `PctTargetSelMTD` (e.g., "43%") centered over the top two rows of circles in the waffle chart.

Below is the revised script, followed by an explanation of the changes and instructions for implementing it in Deneb.

Updated Deneb Vega-Lite JSON Script

```
json

{
  "$schema": "https://vega.github.io/schema/vega-lite/v5.json",
  "description": "Dynamic Waffle Chart with Enlarged Circles and Centered Percentage",
  "data": {
    "name": "dataset"
  },
  "transform": [
    {
      "calculate": "(datum.id - 1) % 10",
      "as": "col"
    },
    {
      "calculate": "9 - floor((datum.id - 1) / 10)",
      "as": "row"
    },
    {
      "calculate": "datum.id <= floor(datum.PctTargetSelMTD * 100 + 0.5) ? 'filled' : 'empty'",
      "as": "status"
    }
  ],
  "encoding": {
    "x": {
      "field": "col",
      "type": "ordinal",
      "axis": null,
      "scale": {
        "paddingInner": 0.3
      }
    },
    "y": {
      "field": "row",
      "type": "ordinal",
      "axis": null,
      "scale": {
        "paddingInner": 0.3
      }
    }
  },
  "layer": [
    {
      "mark": {
        "type": "circle",
        "size": 300
      },
      "encoding": {
        "fill": {
          "field": "status",
          "type": "nominal",
          "scale": {
            "domain": ["filled", "empty"],
            "range": ["darkblue", "lightblue"]
          },
          "legend": null
        }
      }
    },
    {
      "mark": {
        "type": "text",
        "align": "center",
        "baseline": "bottom",
        "fontSize": 24,
        "fontWeight": "bold",
        "dy": -10
      },
      "encoding": {
        "text": {
          "field": "PctTargetSelMTD",
          "type": "quantitative",
        }
      }
    }
  ]
}
```

```
        "format": ".0%",
        "aggregate": "mean"
    },
    "x": {
        "aggregate": "mean",
        "field": "col"
    },
    "y": {
        "value": -1
    }
}
],
"title": {
    "text": "Percentage of Annual Sales Target Achieved",
    "fontSize": 16,
    "fontWeight": "bold"
},
"config": {
    "view": {
        "stroke": null
    }
}
}
```

Explanation of Changes

1. Increased Circle Size by 50%

- Circle Size Increased from 200 to 300:

json

```
"mark": {
    "type": "circle",
    "size": 300
},
```

- The `size` property of the circle mark has been increased by 50% (from 200 to 300) to make the circles larger.

2. Added More Spacing Between Circles

- Adjusted `paddingInner` in `x` and `y` Encodings:

json

```
"scale": {
    "paddingInner": 0.3
}
```

- Increased `paddingInner` from the default to `0.3` to add more spacing between the circles. This value can be adjusted as needed.

3. Displayed PctTargetSelMTD Centered Over the Top Two Rows

- Added a New Layer with a Text Mark:

json

```
{
    "mark": {
        "type": "text",
        "align": "center",
        "baseline": "bottom",
        "fontSize": 24,
        "fontWeight": "bold",
        "dy": -10
    },
    "encoding": {
        "text": {
            "field": "PctTargetSelMTD",
            "type": "quantitative",
            "format": ".0%",
            "aggregate": "mean"
        }
    }
}
```

```
        "x": {
          "aggregate": "mean",
          "field": "col"
        },
        "y": {
          "value": -1
        }
      }
    }
```

- Positioning the Text:
 - **x** Encoding:
 - Positioned horizontally at the center of the waffle chart by taking the mean of the `col` field.
 - **y** Encoding:
 - Positioned vertically above the top row with `"value": -1`.
 - **dy**:
 - Adjusted the vertical offset with `"dy": -10` to fine-tune the placement over the top two rows.
- Displaying the Percentage:
 - **text** Encoding:
 - Displays the `PctTargetSelMTD` value formatted as a percentage with no decimal places (`format: ".0%"`).
 - **aggregate**:
 - Used `"aggregate": "mean"` to ensure a single value is displayed.

4. Layered the Marks

- Used the `layer` Property:

json

```
"layer": [
  { ... }, // Circle marks
  { ... } // Text mark
]
```

- The circle marks and the text mark are placed in separate layers to overlay them correctly.

5. Shared Encodings

- Moved **x** and **y** Encodings to the Top Level:

json

```
"encoding": {
  "x": { ... },
  "y": { ... }
}
```

- Placed **x** and **y** encodings at the top level to be shared across layers.

Instructions to Implement the Script in Deneb

1. Prepare the Data

- Create the `WaffleData` Table in Power BI:

DAX

```
WaffleData =
GENERATESERIES(1, 100, 1)
```

- This table will have a single column `Value` with values from 1 to 100.
- Rename the `Value` Column to `id`:

DAX

```
id = [Value]
```

2. Add Fields and Measures to Deneb

- Add `id` from `WaffleData`:
 - Drag the `id` field into the Deneb visual's **Fields** pane.
- Add `PctTargetSelMTD` Measure:
 - Drag the `PctTargetSelMTD` measure into the Deneb visual's **Fields** pane.
- Deneb's Combined Dataset:
 - Deneb will combine these fields into a single dataset named `dataset`.

3. Verify Field Names

- Ensure that the field names in the script (`id, PctTargetSelMTD`) exactly match those in Deneb, including case sensitivity.

4. Implement the Updated Script

- Copy the Revised Script:
 - Paste the updated JSON script into the Deneb visual's specification.

5. Test the Visual

- Check the Circle Sizes and Spacing:
 - Verify that the circles are larger and have more spacing between them.
- Verify the Percentage Display:
 - Ensure that the `PctTargetSelMTD` value (e.g., "43%") is displayed centered over the top two rows of circles.
- Adjust if Necessary:
 - If the text is not perfectly centered or overlapping the desired rows, adjust the "`dy`" value in the text mark's `mark` property.

json

```
"dy": -10 // Adjust this value as needed
```

Additional Notes

- Interactivity Remains:
 - The waffle chart will continue to update dynamically based on changes to the `PctTargetSelMTD` measure.
- Fine-Tuning Spacing and Positioning:
 - Circle Spacing:
 - Adjust the `paddingInner` values in the `x` and `y` encodings to increase or decrease spacing between circles.

json

```
"paddingInner": 0.3 // Increase for more spacing, decrease for less
```

- Text Positioning:
 - Modify the "`dy`" property in the text mark to fine-tune vertical positioning.
 - Adjust the "`y`" value if needed to move the text higher or lower.
- Formatting the Percentage:
 - The `format: ".0%"` in the `text` encoding ensures that the percentage is displayed without decimal places.

Example DAX Measure for `PctTargetSelMTD` (If Needed)

DAX

```
PctTargetSelMTD =
DIVIDE(
    [Sales Current YTD],
    [Current Year Sales Target],
```

```
    0  
)
```

- Ensure that this measure calculates the percentage of the annual sales target achieved to date and returns a decimal between 0 and 1.

Troubleshooting Tips

- Circles Not Displaying Correctly:
 - Verify that the `id` field contains values from 1 to 100.
 - Ensure that `PctTargetSelMTD` returns a decimal between 0 and 1.
- Percentage Not Displaying or Incorrect:
 - Check that `PctTargetSelMTD` is properly added to the Deneb visual's Fields pane.
 - Ensure that field names match exactly in the script and Deneb.
- Adjusting for Different Screen Sizes:
 - If the visual doesn't appear correctly on different screen sizes, you may need to adjust the circle `size` and `paddingInner` values accordingly.

Final Thoughts

By making these adjustments, the waffle chart should now have larger circles with increased spacing and display the percentage value centered over the top two rows. This enhances the visual appeal and provides immediate insight into the percentage of the annual sales target achieved.

If you have any further questions or need additional assistance fine-tuning the chart, please don't hesitate to ask. I'm here to help you get your visual exactly as you envision it!

Getting closer, but now 2 identical waffles appear side-by-side

```
{  
  "$schema": "https://vega.github.io/schema/vega-lite/v5.json",  
  "description": "Dynamic Waffle Chart with Enlarged Circles and Centered  
    Percentage",  
  "data": {  
    "name": "dataset"  
  },  
  "transform": [  
    {  
      "calculate": "(datum.id - 1) % 10",  
      "as": "col"  
    },  
    {  
      "calculate": "9 - floor((datum.id - 1) / 10)",  
      "as": "row"  
    },  
    {  
      "calculate": "datum.id <= floor(datum.PctTargetSelMTD * 100 + 0.5) ? 'filled' :  
        'empty'",  
      "as": "status"  
    }  
  ],  
  "encoding": {  
    "x": {
```

```
        "field": "col",
        "type": "ordinal",
        "axis": null,
        "scale": {
          "paddingInner": 0.3
        }
      },
      "y": {
        "field": "row",
        "type": "ordinal",
        "axis": null,
        "scale": {
          "paddingInner": 0.3
        }
      }
    },
    "layer": [
      {
        "mark": {
          "type": "circle",
          "size": 300
        },
        "encoding": {
          "fill": {
            "field": "status",
            "type": "nominal",
            "scale": {
              "domain": ["filled", "empty"],
              "range": ["darkblue", "lightblue"]
            }
          },
          "legend": null
        }
      },
      {
        "mark": {
          "type": "text",
          "align": "center",
          "baseline": "bottom",
          "fontSize": 24,
          "fontWeight": "bold",
          "dy": -10
        },
        "encoding": {
          "text": {
            "field": "PctTargetSelMTD",
            "type": "quantitative",
            "format": ".0%",
            "aggregate": "mean"
          },
          "x": {
            "aggregate": "mean",
            "field": "col"
          }
        }
      }
    ]
  }
}
```

```

        "y": {
          "value": -1
        }
      }
    }
  ],
  "title": {
    "text": "Percentage of Annual Sales Target Achieved",
    "fontSize": 16,
    "fontWeight": "bold"
  },
  "config": {
    "view": {
      "stroke": null
    }
  }
}

```

I understand that you're seeing two identical waffle charts side-by-side, and you'd like to resolve this issue so only one chart appears. This problem is likely due to how the data is being combined in Deneb, resulting in duplicated data that causes two charts to render.

Cause of the Issue

When you add both the `id` field (from your `WaffleData` table) and the `PctTargetSelMTD` measure to the Deneb visual's Fields pane, Deneb combines these into a dataset where each `id` is associated with each instance of `PctTargetSelMTD`. This can lead to data duplication, effectively doubling the number of rows, and causing two identical waffle charts to appear side-by-side.

Solution

To fix this, we need to adjust the Vega-Lite specification to prevent data duplication:

- ➊ Use a `joinaggregate` transform to attach the mean (since it's a single value) of `PctTargetSelMTD` to each `id` without duplicating rows.
- ➋ Ensure the `id` field remains unique and that the measure is properly aggregated.

Updated Deneb Vega-Lite JSON Script

```

json

{
  "$schema": "https://vega.github.io/schema/vega-lite/v5.json",
  "description": "Dynamic Waffle Chart with Enlarged Circles and Centered Percentage",
  "data": {
    "name": "dataset"
  },
  "transform": [
    {
      "calculate": "(datum.id - 1) % 10",
      "as": "col"
    },
    {
      "calculate": "9 - floor((datum.id - 1) / 10)",
      "as": "row"
    },
    {
      "joinaggregate": [
        {
          "op": "mean",

```

```
        "field": "PctTargetSelMTD",
        "as": "PctTargetSelMTDValue"
    }
],
{
    "calculate": "datum.id <= floor(datum.PctTargetSelMTDValue * 100 + 0.5) ? 'filled' : 'empty'",
    "as": "status"
},
],
"encoding": {
    "x": {
        "field": "col",
        "type": "ordinal",
        "axis": null,
        "scale": {
            "paddingInner": 0.3
        }
    },
    "y": {
        "field": "row",
        "type": "ordinal",
        "axis": null,
        "scale": {
            "paddingInner": 0.3
        }
    }
},
"layer": [
    {
        "mark": {
            "type": "circle",
            "size": 300
        },
        "encoding": {
            "fill": {
                "field": "status",
                "type": "nominal",
                "scale": {
                    "domain": ["filled", "empty"],
                    "range": ["darkblue", "lightblue"]
                },
                "legend": null
            }
        }
    },
    {
        "mark": {
            "type": "text",
            "align": "center",
            "baseline": "bottom",
            "fontSize": 24,
            "fontWeight": "bold",
            "dy": -10
        },
        "encoding": {
            "text": {
                "field": "PctTargetSelMTDValue",
                "type": "quantitative",
                "format": ".0%"
            },
            "x": {
                "aggregate": "mean",
                "field": "col"
            },
            "y": {
                "value": -1
            }
        }
    ],
    "title": {
        "text": "Percentage of Annual Sales Target Achieved",
        "fontSize": 16,
        "fontWeight": "bold"
    },
    "config": {
        "view": {
```

```
        "stroke": null
    }
}
```

Explanation of Changes

1. Added joinaggregate Transform

- Purpose: Attaches the mean value of `PctTargetSelMTD` to each data row without causing duplication.
- Implementation:

json

```
{
  "joinaggregate": [
    {
      "op": "mean",
      "field": "PctTargetSelMTD",
      "as": "PctTargetSelMTDValue"
    }
  ]
}
```

- Effect: The `PctTargetSelMTDValue` field now contains the same value for all rows, preventing duplication of the `id` field.

2. Updated Calculations to Use `PctTargetSelMTDValue`

- Status Calculation:

json

```
{
  "calculate": "datum.id <= floor(datum.PctTargetSelMTDValue * 100 + 0.5) ? 'filled' : 'empty'",
  "as": "status"
}
```

- Text Encoding:

json

```
{
  "text": {
    "field": "PctTargetSelMTDValue",
    "type": "quantitative",
    "format": ".0%"
  }
}
```

3. Ensured Unique `id` Field

- By using the `joinaggregate` transform, we prevent duplication of the `id` field, ensuring each `id` is unique and only one waffle chart is rendered.

Instructions to Implement the Script in Deneb

1. Prepare the Data

- Create `WaffleData` Table in Power BI:

DAX

```
WaffleData =
ADDCOLUMNS(
    GENERATESERIES(1, 100, 1),
    "id", [Value]
)
```

- This table has a single column `id` with values from 1 to 100.
- Ensure `WaffleData` only contains the `id` column.

2. Add Fields and Measures to Deneb

- Add `id` from `WaffleData`:
 - Drag the `id` field into the Deneb visual's **Fields** pane.
- Add `PctTargetSelMTD` Measure:
 - Drag the `PctTargetSelMTD` measure into the Deneb visual's **Fields** pane.
- Deneb's Combined Dataset:
 - Deneb will combine these into a single dataset named `dataset`.

3. Verify Field Names

- Ensure the field names in the script (`id, PctTargetSelMTD`) match exactly with those in Deneb, including case sensitivity.

4. Implement the Updated Script

- Copy the Revised Script:
 - Paste the updated JSON script into the Deneb visual's specification.

5. Test the Visual

- Check for Duplication:
 - Verify that only one waffle chart appears.
- Verify the Percentage Display:
 - Ensure the `PctTargetSelMTD` value (e.g., "43%") is displayed centered over the top rows of the chart.

Additional Notes

- Interactivity Remains:
 - The chart will continue to update dynamically based on changes to the `PctTargetSelMTD` measure.
- Fine-Tuning Spacing and Positioning:
 - Circle Spacing:
 - Adjust the `paddingInner` values in the `x` and `y` encodings to modify spacing between circles.
 - Text Positioning:
 - Modify the "`dy`" property in the text mark to fine-tune vertical positioning.

Troubleshooting Tips

- If Duplication Persists:
 - Ensure the `joinaggregate` transform is correctly included in the script.
 - Confirm that `PctTargetSelMTD` is added as a measure, not as a column in the same table as `id`.
- Data Field Names:
 - Double-check that all field names in the script match those in Deneb exactly, including case sensitivity.
- Circles Not Displaying Correctly:
 - Verify that the `id` field contains values from 1 to 100 and is unique.
 - Ensure `PctTargetSelMTD` returns a decimal between 0 and 1.

Final Thoughts

By adjusting the script to include the `joinaggregate` transform, we've prevented data duplication caused by the combination of the `id` field and the `PctTargetSelMTD` measure. This ensures that only one waffle chart is displayed, as intended, with the percentage value centered over the top rows.

If you have any further questions or need additional assistance fine-tuning the chart, please don't hesitate to ask. I'm here to help you get your visual exactly as you envision it!

Example DAX Measure for `PctTargetSelMTD` (If Needed)

DAX

```
PctTargetSelMTD =  
DIVIDE(  
    [Sales Current YTD],  
    [Current Year Sales Target],  
    0  
)
```

- Ensure this measure calculates the percentage of the annual sales target achieved to date and returns a decimal between 0 and 1.

Additional Resources

- Vega-Lite Documentation on [joinaggregate Transform](#):
 - Join Aggregate Transform
- Deneb Documentation:
 - [Deneb User Guide](#)

I hope this helps resolve the issue and gets your waffle chart displaying as desired. Let me know if you need any more assistance!

OK, some strange things happened:

No graphics at all showing up now.

Log in Deneb shows this text

[Error] Unexpected token ILLEGAL

[Error] Invalid specification {"datasets":{"dataset":
[{"PctTargetSelMTD":0.2471941550000001,"PctTargetSelMTD_format":"0%;-0
%;0%","PctTargetSelMTD_formatted":"25%","_identity_":
{"dataMap":null,"deepestNodeLevelInPath":-1,"key":null["Sum(WaffleData.id)\\\"\\"
"Measures.PctTargetSelMTD\\\"]","measures":["Sum(WaffleData.id)","
"Measures.PctTargetSelMTD"],"nodeIndexes":[],"nodeIsCollapsed":null};"_key_":
{"metadata":null,"row":0,"id":5050,"id_format":0,"id_f
ormatted":5050}]}]. Make sure the specification includes at least one of the
following properties: "mark", "layer", "facet", "hconcat", "vconcat", "concat", or
"repeat".

In the field mapping, PctTargetSelMTD shows up twice, while id shows up correctly
(i.e. once_

I'm sorry to hear that you're experiencing issues with the latest script. Let's work together to identify and resolve the problem so that your waffle chart displays correctly.

Understanding the Issue

From your message, here's what we know:

- No graphics are showing up in the visual.
- Deneb Log Error Messages:

csharp

```
[Error] Unexpected token ILLEGAL  
[Error] Invalid specification {"datasets":{"dataset":[]}}. Make sure the specification includ
```



- `PctTargetSelMTD` shows up twice in the field mapping, while `id` shows up once.

These error messages suggest that there's an issue with the data being passed to the Vega-Lite specification and possibly with the field mappings in Deneb.

Possible Causes

1 Data Duplication or Incorrect Data Structure:

- 2 The `dataset` contains only one data point instead of 100, as evidenced by the `id` value of 5050.
- 3 This suggests that the data isn't being passed correctly to the visual.

4 Field Mapping Issues:

- 5 `PctTargetSelMTD` appearing twice in the field mappings could indicate that it's being added both as a measure and as a column, causing conflicts.

6 Incorrect Field Names or Data Types:

- 7 Mismatched field names or data types between the script and the actual data could cause errors.

8 Syntax Errors in the Script:

- 9 There might be syntax issues in the JSON script causing it to be invalid.

Steps to Resolve the Issue

1. Verify the Data Source

Ensure `WaffleData` Table is Correctly Created:

DAX Formula for `WaffleData`:

DAX

```
WaffleData =  
GENERATESERIES(1, 100, 1)
```

Add a Calculated Column Named `id`:

DAX

```
id = [Value]
```

Ensure `WaffleData` Contains 100 Rows with `id` from 1 to 100.

2. Check Field Mappings in Deneb

Remove Any Duplicate Fields:

Ensure `PctTargetSelMTD` Only Appears Once in the Fields Pane:

- It should be added as a measure, not as a column in `WaffleData`.

Verify `id` Field:

- Ensure that the `id` field from `WaffleData` is added correctly.

Clean Up Fields Pane:

- Remove any unintended fields or measures that may have been added.

3. Adjust the Vega-Lite Specification

Use `transform` to Generate `id` if Necessary:

- If the `id` field isn't correctly populated, we can generate it within the Vega-Lite specification.

Revised Script:


```

        "text": {
          "field": "PctTargetSelMTDValue",
          "type": "quantitative",
          "format": ".0%"
        },
        "x": {
          "aggregate": "mean",
          "field": "col"
        },
        "y": {
          "value": -1
        }
      }
    ],
    "title": {
      "text": "Percentage of Annual Sales Target Achieved",
      "fontSize": 16,
      "fontWeight": "bold"
    },
    "config": {
      "view": {
        "stroke": null
      }
    }
  }
}

```

Notes:

- Used `window` Transform to Generate `id`:
 - This ensures that `id` is generated within the specification if the field isn't correctly passed from the data.
- Removed Direct Reference to `id` Field from Data:
 - This can help if there's an issue with the `id` field in the dataset.

4. Re-Add Fields in Deneb

Start Fresh with the Fields Pane:

- ① Remove All Fields from the Fields Pane in Deneb.
- ② Add `PctTargetSelMTD` as a Measure:
 - ③ Ensure it's added only once.
- ④ Do Not Add Any Other Fields:
 - ⑤ Since we're generating `id` within the Vega-Lite specification, we don't need to add the `id` field from `WaffleData`.

Deneb's Dataset Will Now Contain Only `PctTargetSelMTD`.

5. Test the Visual

- Verify That the Waffle Chart Now Displays Correctly.
- Ensure Only One Waffle Chart Appears.
- Check That the Percentage Value is Displayed Correctly Above the Chart.

Additional Explanation

Why Remove the `id` Field from the Data:

- Since adding `id` from `WaffleData` may be causing duplication or incorrect data structure, generating `id` within the Vega-Lite specification can circumvent this issue.

Using `window` Transform to Generate `id`:

- The `window` transform can create a sequential `id` for each data point, ensuring we have 100 data points for the waffle chart.

Ensuring `PctTargetSelMTD` is Correctly Used:

- By aggregating `PctTargetSelMTD` using `joinaggregate`, we can attach its value to each generated data point without causing duplication.

Potential Issues and Solutions

Issue: No Data Points in the Dataset

- Cause: If only `PctTargetSelMTD` is in the dataset and we don't generate `id`, there will be only one data point.
- Solution: Use the `window` transform to generate 100 `id` values within the Vega-Lite specification.

Issue: `PctTargetSelMTD` Not Accessible

- Cause: If `PctTargetSelMTD` is not correctly added as a measure.
- Solution: Ensure `PctTargetSelMTD` is added to the Fields pane as a measure, and it's accessible in the specification.

Issue: Syntax Errors in the Script

- Cause: Missing commas, incorrect JSON formatting, or invalid property names.
- Solution: Carefully check the JSON syntax. Use a JSON validator if necessary.

Alternate Approach

If the above solution doesn't resolve the issue, here's an alternative:

Create a Calculated Table in Power BI

- DAX Formula:

DAX

```
WaffleData =
VAR PercentFilled = [PctTargetSelMTD]
VAR FilledCount = ROUND(PercentFilled * 100, 0)
RETURN
GENERATE(
    GENERATESERIES(1, 100, 1),
    ROW(
        "id", [Value],
        "status", IF([Value] <= FilledCount, "filled", "empty")
    )
)
```

- This Table Will Contain:

- `id` from 1 to 100.
- `status` indicating "filled" or "empty" based on `PctTargetSelMTD`.

Note: This approach embeds the logic within Power BI, simplifying the Vega-Lite specification.

Adjust the Vega-Lite Specification

- Simplified Script:

json

```
{
  "$schema": "https://vega.github.io/schema/vega-lite/v5.json",
  "description": "Simplified Waffle Chart",
  "data": {
    "name": "dataset"
  },
  "transform": [
    {
      "calculate": "(datum.id - 1) % 10",
      "as": "col"
    },
    {
      "calculate": "9 - floor((datum.id - 1) / 10)",
      "as": "row"
    }
  ],
  "mark": {
    "type": "circle",
    "size": 300
  },
  "encoding": {
    "x": {
```

```

    "field": "col",
    "type": "ordinal",
    "axis": null,
    "scale": {
      "paddingInner": 0.3
    }
  },
  "y": {
    "field": "row",
    "type": "ordinal",
    "axis": null,
    "scale": {
      "paddingInner": 0.3
    }
  },
  "fill": {
    "field": "status",
    "type": "nominal",
    "scale": {
      "domain": ["filled", "empty"],
      "range": ["darkblue", "lightblue"]
    },
    "legend": null
  }
},
"title": {
  "text": "Percentage of Annual Sales Target Achieved",
  "fontSize": 16,
  "fontWeight": "bold"
},
"config": {
  "view": {
    "stroke": null
  }
}
}

```

- **Implementation Steps:**

- Add `id` and `status` from `WaffleData` to the Deneb Fields pane.
- Do not add `PctTargetSelMTD` directly; it's already used in the calculated table.

Final Recommendations

- Double-Check All Field Names and Data Types:
 - Ensure consistency between the data and the Vega-Lite specification.
- Use a JSON Validator:
 - Validate your Vega-Lite JSON script using an online tool to catch any syntax errors.
- Consult Deneb Documentation:
 - Refer to the [Deneb User Guide](#) for guidance on field mappings and data handling.

Summary

- **Issue Identified:** Data duplication and incorrect data structure causing the visual not to render.
- **Solution Provided:**
 - Adjusted the Vega-Lite script to generate `id` within the specification.
 - Simplified the data requirements by only needing `PctTargetSelMTD` as a measure.
 - Offered an alternative approach by creating a calculated table in Power BI.
- **Next Steps:**
 - Implement the revised script.
 - Clean up the Fields pane in Deneb.
 - Test the visual and adjust as necessary.

If you continue to experience issues, please provide:

- A screenshot of your Deneb Fields pane.
- Any additional error messages from Deneb.
- Confirmation of the data structure in `WaffleData` and how `PctTargetSelMTD` is added.

I'm here to help you get your waffle chart working perfectly. Let me know how it goes!

OK, back on track. We're getting one 10 x 10 waffle colored appropriately. Here are the issues we still need to resolve:

1. there's now an x axis on the chart entitled "Mean of col" which shouldn't be there
2. PctTargetSelMTD callout in the visual is right justified instead of centered
3. Circles are still not the proper size, but this could be due to the presence of the x axis

OK, back on track. We're getting one 10 x 10 waffle colored appropriately. Here are the issues we still need to resolve:

1. there's now an x axis on the chart entitled "Mean of col" which shouldn't be there
2. PctTargetSelMTD callout in the visual is right justified instead of centered
3. Circles are still not the proper size, but this could be due to the presence of the x axis

Here's the current specification:

```
{  
  "$schema": "https://vega.github.io/schema/vega-lite/v5.json",  
  "description": "Dynamic Waffle Chart",  
  "data": {  
    "name": "dataset"  
  },  
  "transform": [  
    {  
      "window": [{"op": "row_number", "as": "id"}]  
    },  
    {  
      "calculate": "(datum.id - 1) % 10",  
      "as": "col"  
    },  
    {  
      "calculate": "((id - 1) / 10) % 10",  
      "as": "row"  
    }  
  ],  
  "mark": "circle",  
  "encoding": {  
    "color": "value",  
    "x": "col",  
    "y": "row",  
    "size": 1000  
  },  
  "layer": [  
    {  
      "mark": "rule",  
      "encoding": {  
        "x": "col",  
        "x2": "col",  
        "y": "row",  
        "y2": "row + 1"  
      }  
    },  
    {  
      "mark": "rule",  
      "encoding": {  
        "x": "col - 1",  
        "x2": "col + 1",  
        "y": "row",  
        "y2": "row"  
      }  
    }  
  ]  
}
```

```
"calculate": "9 - floor((datum.id - 1) / 10)",
            "as": "row"
        },
        {
            "joinaggregate": [
                {
                    "op": "mean",
                    "field": "PctTargetSelMTD",
                    "as": "PctTargetSelMTDValue"
                }
            ]
        },
        {
            "calculate": "datum.id <= floor(datum.PctTargetSelMTDValue * 100 + 0.5) ? 'filled' :
                'empty'",
            "as": "status"
        }
    ],
    "layer": [
        {
            "mark": {
                "type": "circle",
                "size": 300
            },
            "encoding": {
                "x": {
                    "field": "col",
                    "type": "ordinal",
                    "axis": null,
                    "scale": {
                        "paddingInner": 0.1
                    }
                },
                "y": {
                    "field": "row",
                    "type": "ordinal",
                    "axis": null,
                    "scale": {
                        "paddingInner": 0.1
                    }
                },
                "fill": {
                    "field": "status",
                    "type": "nominal",
                    "scale": {
                        "domain": ["filled", "empty"],
                        "range": ["darkblue", "lightblue"]
                    }
                },
                "legend": null
            }
        }
    ],
    {
        "mark": {
```

```
        "type": "text",
        "align": "center",
        "baseline": "bottom",
        "fontSize": 24,
        "fontWeight": "bold",
        "dy": -10
      },
      "encoding": {
        "text": {
          "field": "PctTargetSelMTDValue",
          "type": "quantitative",
          "format": ".0%"
        },
        "x": {
          "aggregate": "mean",
          "field": "col"
        },
        "y": {
          "value": -1
        }
      },
      "title": {
        "text": "Percentage of Annual Sales Target Achieved",
        "fontSize": 16,
        "fontWeight": "bold"
      },
      "config": {
        "view": {
          "stroke": null
        }
      }
    }
  }
}
```

Hello,

I'm glad to hear that we're back on track and that the waffle chart is displaying correctly with the appropriate coloring. Let's address each of the issues you've listed one by one.

Issue 1: Unwanted x-axis titled "Mean of col"

Cause:

The x-axis is appearing because in the `text mark layer`, we've specified an x-encoding with an aggregate function but didn't suppress the axis for that encoding. Since the x-encoding in the text layer is creating its own axis, it's showing up on the chart.

Solution:

To remove the unwanted x-axis, we need to explicitly set `"axis": null` in the x-encoding of the `text mark layer`. Additionally, to ensure consistent axis settings across layers, we can move the x and y encodings to the `top level` so that they are shared across all layers.

Issue 2: PctTargetSelMTD callout is right-justified instead of centered

Cause:

Even though we've set "align": "center" in the text mark, the positioning may not be centered due to how the x-value is being calculated. Using "aggregate": "mean" on the "col" field might not position the text exactly at the center of the chart.

Solution:

Since our columns (`col` values) range from 0 to 9, the center of the chart horizontally is at $(0 + 9) / 2 = 4.5$. We can set the x-position of the text mark to this value to center the text.

Issue 3: Circles are not the proper size

Cause:

This might be due to the unwanted x-axis affecting the layout, or the circle size may need adjustment after resolving the axis issue.

Solution:

Once we remove the x-axis, the layout should adjust accordingly. Additionally, we can increase the circle "size" property to make the circles larger.

Updated Deneb Vega-Lite JSON Script

Here's the revised script with the necessary changes:

```
json

{
  "$schema": "https://vega.github.io/schema/vega-lite/v5.json",
  "description": "Dynamic Waffle Chart",
  "data": {
    "name": "dataset"
  },
  "transform": [
    {
      "window": [{"op": "row_number", "as": "id"}]
    },
    {
      "calculate": "(datum.id - 1) % 10",
      "as": "col"
    },
    {
      "calculate": "9 - floor((datum.id - 1) / 10)",
      "as": "row"
    },
    {
      "joinaggregate": [
        {
          "op": "mean",
          "field": "PctTargetSelMTD",
          "as": "PctTargetSelMTDValue"
        }
      ]
    },
    {
      "calculate": "datum.id <= floor(datum.PctTargetSelMTDValue * 100 + 0.5) ? 'filled' : 'empty'",
      "as": "status"
    }
  ],
  "encoding": {
    "x": {
      "field": "col",
      "type": "ordinal",
      "axis": null,
      "scale": {
        "paddingInner": 0.1
      }
    },
    "y": {
      "field": "row",
      "type": "ordinal",
      "axis": null,
      "scale": {
        "paddingInner": 0.1
      }
    }
  }
}
```

```

        },
      },
      "layer": [
        {
          "mark": {
            "type": "circle",
            "size": 500
          },
          "encoding": {
            "fill": {
              "field": "status",
              "type": "nominal",
              "scale": {
                "domain": ["filled", "empty"],
                "range": ["darkblue", "lightblue"]
              },
              "legend": null
            }
          }
        },
        {
          "mark": {
            "type": "text",
            "align": "center",
            "baseline": "bottom",
            "fontSize": 24,
            "fontWeight": "bold",
            "dy": -10
          },
          "encoding": {
            "text": {
              "field": "PctTargetSelMTDValue",
              "type": "quantitative",
              "format": ".0%"
            },
            "x": {
              "value": 4.5,
              "axis": null
            },
            "y": {
              "value": -1
            }
          }
        }
      ],
      "title": {
        "text": "Percentage of Annual Sales Target Achieved",
        "fontSize": 16,
        "fontWeight": "bold"
      },
      "config": {
        "view": {
          "stroke": null
        }
      }
    }
  ]
}

```

Explanation of Changes

1. Moved **x** and **y** Encodings to the Top Level

- Purpose: To share the x and y encodings across all layers and ensure consistent axis settings.
- Changes:

json

```

"encoding": {
  "x": {
    "field": "col",
    "type": "ordinal",
    "axis": null,
    "scale": {
      "paddingInner": 0.1
    }
  },
  "y": {
    "field": "row",
    "type": "nominal",
    "axis": null,
    "scale": {
      "paddingOuter": 0.1
    }
  }
}

```

```
        "field": "row",
        "type": "ordinal",
        "axis": null,
        "scale": {
          "paddingInner": 0.1
        }
      },
    },
```

- Effect: This removes any axes from appearing in the chart and applies the same x and y positioning to both the circle and text marks.

2. Adjusted Text Mark Positioning

- Set `x` Position to Center:

json

```
"x": {
  "value": 4.5,
  "axis": null
}
```

- Explanation: By setting `x` to 4.5 (the center of the 0-9 range), the text is horizontally centered.
- Removed Aggregate Function: No need for `"aggregate": "mean"` since we're specifying an exact value.
- Set `"axis": null`: To ensure no axis is created due to this encoding.

- Set `y` Position:

json

```
"y": {
  "value": -1
}
```

- Explanation: Positions the text above the top row of the waffle chart.

3. Increased Circle Size

- Adjusted Circle Size:

json

```
"size": 500
```

- Explanation: Increased from 300 to 500 to make the circles larger. You can adjust this value as needed.

4. Ensured No Axes Appear

- Set `"axis": null` in All Encodings:

- Both in the top-level `x` and `y` encodings and within the text mark's `x` encoding.

Instructions to Implement the Changes

1. Update the Vega-Lite Specification in Deneb

- Replace your current JSON code with the updated script provided above.

2. Verify Field Mappings

- Ensure `PctTargetSelMTD` is Correctly Added:

- Add `PctTargetSelMTD` as a measure in Deneb's **Fields** pane.

- No Additional Fields Needed:

- Since we're generating `id` within the Vega-Lite specification, you don't need to add any other fields.

3. Adjust Circle Size if Necessary

- Modify the Circle `size` Property:

json

```
"size": 500 // Increase or decrease this value as needed
```

4. Fine-Tune Text Positioning

- Adjust `x` Value for Horizontal Positioning:

json

```
"x": {  
  "value": 4.5,  
  "axis": null  
}
```

- Explanation: If the text isn't perfectly centered, you can adjust the `value` slightly (e.g., 4.4 or 4.6).

- Adjust `dy` for Vertical Positioning:

json

```
"dy": -10 // Increase (more negative) to move text up, decrease to move it down
```

Troubleshooting Tips

- Circles Not Displaying Correctly:

- Ensure that `PctTargetSelMTD` returns a decimal between 0 and 1.
- Verify that `id` is being generated correctly via the `window` transform.

- Text Not Centered:

- Adjust the `x` encoding's `value` in the text mark.
- Make sure the `align` property is set to "center".

- Axes Still Appearing:

- Confirm that `"axis": null` is set in all x and y encodings.
- Ensure there are no typos in the property names.

Final Thoughts

By making these adjustments, we should have resolved all three issues:

- 1 Removed the Unwanted x-axis:

- 2 By moving x and y encodings to the top level and setting `"axis": null`, we've prevented axes from appearing.

- 3 Centered the `PctTargetSelMTD` Callout:

- 4 By setting the `x` value to 4.5 in the text mark's encoding, the percentage should now be horizontally centered.

- 5 Adjusted Circle Sizes:

- 6 Increasing the circle `size` and removing the axes should result in circles displaying at the desired size.

Example DAX Measure for `PctTargetSelMTD` (If Needed)

DAX

```
PctTargetSelMTD =  
DIVIDE(  
  [Sales Current YTD],  
  [Current Year Sales Target],  
  0  
)
```

- Ensure that this measure returns a decimal between 0 and 1.

Need Further Assistance?

If you have any further questions or need additional help fine-tuning the chart, please don't hesitate to ask. I'm here to help you get your visual exactly as you envision it!

OK, we're very close now. Just need to resolve a few minor issues:

1. Add a space after the title
2. Decrease the space between columns of the waffle

I'm glad to hear we're almost there! Let's address the two minor issues you've mentioned:

- 1 Add a space after the title.
- 2 Decrease the space between columns of the waffle.

Issue 1: Add a Space After the Title

Solution:

To add space after the title, we can adjust the `offset` property in the `title` specification. The `offset` property controls the space in pixels between the title and the chart.

Issue 2: Decrease the Space Between Columns of the Waffle

Solution:

To decrease the space between the columns of the waffle chart, we need to adjust the `paddingInner` property in the `x` encoding's `scale`. The `paddingInner` property controls the spacing between the bands in a band scale. Reducing this value will decrease the space between columns.

Updated Deneb Vega-Lite JSON Script

Here's the revised script with the necessary adjustments:

```
json

{
  "$schema": "https://vega.github.io/schema/vega-lite/v5.json",
  "description": "Dynamic Waffle Chart",
  "data": {
    "name": "dataset"
  },
  "transform": [
    {
      "window": [{"op": "row_number", "as": "id"}]
    },
    {
      "calculate": "(datum.id - 1) % 10",
      "as": "col"
    },
    {
      "calculate": "9 - floor((datum.id - 1) / 10)",
      "as": "row"
    },
    {
      "joinaggregate": [
        {
          "op": "mean",
          "field": "PctTargetSelMTD",
          "as": "PctTargetSelMTDValue"
        }
      ]
    }
  ],
  "mark": "waffle",
  "encoding": {
    "x": {
      "band": "col"
    },
    "y": {
      "band": "row"
    },
    "color": {
      "value": "#f0f0f0"
    }
  }
}
```



```
}
```

Explanation of Changes

1. Added Space After the Title

- Adjusted the `offset` Property in the Title:

```
json
```

```
"title": {  
  "text": "Percentage of Annual Sales Target Achieved",  
  "fontSize": 16,  
  "fontWeight": "bold",  
  "offset": 20  
},
```

- Explanation: The default `offset` is 10 pixels. By setting it to `20`, we've added extra space between the title and the chart. You can adjust this value as needed.

2. Decreased Space Between Columns

- Adjusted `paddingInner` in the `x` Encoding's Scale:

```
json
```

```
"x": {  
  "field": "col",  
  "type": "ordinal",  
  "axis": null,  
  "scale": {  
    "paddingInner": 0.05  
  }  
},
```

- Explanation: Reduced `paddingInner` from `0.1` to `0.05` to decrease the space between columns. Lower values result in less space. You can adjust this value to fine-tune the spacing.

- Left `paddingInner` in the `y` Encoding Unchanged:

```
json
```

```
"y": {  
  "field": "row",  
  "type": "ordinal",  
  "axis": null,  
  "scale": {  
    "paddingInner": 0.1  
  }  
},
```

- Explanation: Kept the vertical spacing the same to maintain readability.

Instructions to Implement the Changes

1. Update the Vega-Lite Specification in Deneb

- Replace your current JSON code with the updated script provided above.

2. Verify Field Mappings

- Ensure `PctTargetSelMTD` is Correctly Added:

- Add `PctTargetSelMTD` as a measure in Deneb's Fields pane.

- No Additional Fields Needed:

- Since we're generating `id` within the Vega-Lite specification, you don't need to add any other fields.

3. Adjust Spacing and Sizing if Necessary

- Modify the `offset` in the Title:

json

```
"offset": 20 // Increase for more space, decrease for less
```

- Adjust the `paddingInner` Value in the `x` Encoding:

json

```
"paddingInner": 0.05 // Decrease for less space between columns
```

● Note: Values for `paddingInner` range from `0` (no padding) to `1` (maximum padding). Adjust as needed.

Troubleshooting Tips

- If the Space After the Title Is Not Sufficient:

● Increase the `offset` value in the title to add more space.

- If Columns Are Still Too Spaced Out:

● Further decrease the `paddingInner` value in the `x` encoding.

- If Circles Overlap Due to Reduced Padding:

● Adjust the circle `size` to prevent overlap.

json

```
"size": 500 // Reduce this value if circles overlap
```

- Adjust Vertical Spacing Between Rows (If Needed):

● Modify the `paddingInner` in the `y` encoding's scale.

json

```
"y": {  
    "scale": {  
        "paddingInner": 0.1 // Adjust as needed  
    }  
}
```

Final Thoughts

By making these adjustments, we've:

- 1 Added Space After the Title:

2 The `offset` property in the title adds the desired space between the title and the chart.

- 3 Decreased Space Between Columns:

4 Reducing the `paddingInner` in the `x` encoding's scale decreases the space between columns, making the waffle chart appear denser.

Example DAX Measure for PctTargetSelMTD (If Needed)

DAX

```
PctTargetSelMTD =  
DIVIDE(  
    [Sales Current YTD],  
    [Current Year Sales Target],  
    0  
)
```

- Ensure that this measure returns a decimal between 0 and 1.

Need Further Assistance?

If you have any further questions or need additional help fine-tuning the chart, please don't hesitate to ask. I'm here to help you get your visual exactly as you envision it!

Additional Resources

- Vega-Lite Documentation on Title:
 - Title Properties
- Vega-Lite Documentation on Band Scale and Padding:
 - Scale Padding

Let me know if these adjustments achieve the desired effect or if there's anything else you'd like to modify!

Here

Here's my current specification - the only remaining problem is that I want to move the columns of the waffle closer together, but I've set the x inner padding to 0 and the circle size to 600 and they still don't move closer

```
{  
  "$schema": "https://vega.github.io/schema/vega-lite/v5.json",  
  "description": "Dynamic Waffle Chart",  
  "data": {  
    "name": "dataset"  
  },  
  "transform": [  
    {  
      "window": [{"op": "row_number", "as": "id"}]  
    },  
    {  
      "calculate": "(datum.id - 1) % 10",  
      "as": "col"  
    },  
    {  
      "calculate": "9 - floor((datum.id - 1) / 10)",  
      "as": "row"  
    },  
    {  
      "joinaggregate": [  
        {  
          "op": "mean",  
          "field": "PctTargetSelMTD",  
          "as": "PctTargetSelMTDValue"  
        }  
      ]  
    },  
    {  
      "calculate": "row * 10 + col",  
      "as": "id"  
    }  
  ],  
  "mark": "circle",  
  "encoding": {  
    "x": {"field": "row", "type": "band"},  
    "y": {"field": "col", "type": "band"},  
    "color": {"field": "PctTargetSelMTDValue", "type": "quantitative"},  
    "size": 600  
  }  
}
```

```
"calculate": "datum.id <= floor(datum.PctTargetSelMTDValue * 100 + 0.5) ? 'filled' : 'empty'",  
        "as": "status"  
    },  
    ],  
    "encoding": {  
        "x": {  
            "field": "col",  
            "type": "ordinal",  
            "axis": null,  
            "scale": {  
                "paddingInner": 0  
            }  
        },  
        "y": {  
            "field": "row",  
            "type": "ordinal",  
            "axis": null,  
            "scale": {  
                "paddingInner": 0.05  
            }  
        }  
    },  
    "layer": [  
        {  
            "mark": {  
                "type": "circle",  
                "size": 600  
            },  
            "encoding": {  
                "fill": {  
                    "field": "status",  
                    "type": "nominal",  
                    "scale": {  
                        "domain": ["filled", "empty"],  
                        "range": ["darkblue", "lightblue"]  
                    }  
                }  
            },  
            "legend": null  
        },  
        {  
            "mark": {  
                "type": "text",  
                "align": "center",  
                "baseline": "bottom",  
                "fontSize": 32,  
                "color": "orangered",  
                "fontWeight": "bold",  
                "dy": 40  
            },  
            "encoding": {  
                "text": {  
                    "field": "PctTargetSelMTDValue",  
                    "type": "quantitative",  
                    "format": ".2",  
                    "precision": 2  
                }  
            }  
        }  
    ]  
}
```

```

        "type": "quantitative",
        "format": ".0%"
      },
      "x": {
        "value": 220
      },
      "y": {
        "value": -1
      }
    ],
    "title": {
      "text": "Percentage of Annual Sales Target Achieved",
      "fontSize": 18,
      "fontWeight": "bold",
      "offset": 20
    },
    "config": {
      "view": {
        "stroke": null
      }
    }
  }
}

```

Hello,

I understand that you're trying to reduce the space between the columns of your waffle chart, but even after setting `paddingInner` to `0` and increasing the circle size to `600`, the columns are not moving closer together. This issue occurs because Vega-Lite automatically sizes the chart to fill the available space, distributing the columns evenly across the width. Setting `paddingInner` to `0` alone doesn't force the columns to be closer if the chart width remains large.

Solution:

To bring the columns closer together, you need to **explicitly set the chart's width** to control how much space the columns occupy. By reducing the chart's width, the columns will be forced to move closer together since they have less horizontal space to fill.

Updated Vega-Lite Specification:

Here's your updated specification with the `width` property added and adjustments made to bring the columns closer together:

json

```
{
  "$schema": "https://vega.github.io/schema/vega-lite/v5.json",
  "description": "Dynamic Waffle Chart",
  "width": 300, // Adjust this value as needed
  "height": 300, // Adjust this value as needed
  "data": {
    "name": "dataset"
  },
  "transform": [
    {
      "window": [{"op": "row_number", "as": "id"}]
    },
    {
      "calculate": "(datum.id - 1) % 10",
      "as": "col"
    }
  ],
  "grid": [
    {"x": 0, "y": 0, "x2": 10, "y2": 10}
  ]
}
```

```

},
{
  "calculate": "9 - floor((datum.id - 1) / 10)",
  "as": "row"
},
{
  "joinaggregate": [
    {
      "op": "mean",
      "field": "PctTargetSelMTD",
      "as": "PctTargetSelMTDValue"
    }
  ]
},
{
  "calculate": "datum.id <= floor(datum.PctTargetSelMTDValue * 100 + 0.5) ? 'filled' : 'empty'",
  "as": "status"
},
],
"encoding": {
  "x": {
    "field": "col",
    "type": "ordinal",
    "axis": null,
    "scale": {
      "paddingInner": 0,
      "paddingOuter": 0
    }
  },
  "y": {
    "field": "row",
    "type": "ordinal",
    "axis": null,
    "scale": {
      "paddingInner": 0.05,
      "paddingOuter": 0
    }
  }
},
"layer": [
  {
    "mark": {
      "type": "circle",
      "size": 800 // Increased size for larger circles
    },
    "encoding": {
      "fill": {
        "field": "status",
        "type": "nominal",
        "scale": {
          "domain": ["filled", "empty"],
          "range": ["darkblue", "lightblue"]
        },
        "legend": null
      }
    }
  },
  {
    "mark": {
      "type": "text",
      "align": "center",
      "baseline": "middle",
      "fontSize": 32,
      "color": "orangered",
      "fontWeight": "bold",
      "dy": 0
    },
    "encoding": {
      "text": {
        "field": "PctTargetSelMTDValue",
        "type": "quantitative",
        "format": ".0%"
      },
      "x": {
        "value": 150 // Centered based on the chart width
      },
      "y": {
        "value": -20 // Position above the chart
      }
    }
  }
]
}

```

```

        }
    },
],
"title": {
    "text": "Percentage of Annual Sales Target Achieved",
    "fontSize": 18,
    "fontWeight": "bold",
    "offset": 20
},
"config": {
    "view": {
        "stroke": null
    }
}
}

```

Explanation of Changes:

1. Set the Chart's width and height:

- Added `width` and `height` properties:

json

```

"width": 300,
"height": 300,

```

- Effect:** By setting the chart's width and height to `300` pixels, you're constraining the space in which the columns and rows can be laid out, forcing them to be closer together. You can adjust these values as needed to achieve your desired spacing.

2. Adjusted paddingOuter in the Scales:

- Set `paddingOuter` to `0` in both x and y scales:

json

```

"scale": {
    "paddingInner": 0,
    "paddingOuter": 0
}

```

- Effect:** Setting `paddingOuter` to `0` removes any padding at the ends of the scales, allowing the circles to occupy the full width and height of the chart area.

3. Increased Circle Size:

- Increased the circle `size` to `800`:

json

```

"size": 800

```

- Effect:** With the reduced chart width, increasing the circle size ensures that the circles are as large as possible without overlapping, filling the available space.

4. Adjusted Text Positioning:

- Updated the `x` and `y` values in the text mark's encoding:

json

```

"x": {
    "value": 150 // Half of the chart width to center the text
},
"y": {
    "value": -20 // Adjusted to position the text above the chart
},

```

- Effect:** Centering the text based on the chart's width (`150` is half of `300`) and positioning it appropriately above the chart.

5. Adjusted Text Vertical Alignment:

- Changed `baseline` to `"middle"` and `dy` to `0`:

```
json  
  "baseline": "middle",  
  "dy": 0
```

- **Effect:** Ensures that the text is vertically centered at the specified `y` position.

Fine-Tuning the Chart:

- **Adjust the `width` and `height` Properties:**
 - If you still see space between the columns, try reducing the `width` further or increasing the circle `size`.
- **Adjust Circle `size`:**
 - Be cautious when increasing the circle size to prevent overlap. If the circles start to overlap, reduce the `size` slightly.
- **Experiment with `paddingInner` and `paddingOuter`:**
 - Although you've set `paddingInner` to `0`, if there's still space, ensure that `paddingOuter` is also set to `0`.

Additional Notes:

- **Why the Columns Weren't Moving Closer:**
 - Vega-Lite distributes the ordinal positions across the available space defined by the chart's width and height. If the chart is too wide, even with `paddingInner` set to `0`, the columns will be spread out to fill the space.
- **Controlling Layout with `width` and `height`:**
 - By explicitly setting `width` and `height`, you control the available space for the marks, allowing you to pack them closer together.

Example DAX Measure for `PctTargetSelMTD` (If Needed):

DAX

```
PctTargetSelMTD =  
DIVIDE(  
    [Sales Current YTD],  
    [Current Year Sales Target],  
    0  
)
```

- Ensure that this measure returns a decimal between 0 and 1.

Implementation Steps:

- 1 **Update the Vega-Lite Specification:**
 - 2 Replace your current JSON code with the updated specification provided above.
- 3 **Adjust `width`, `height`, and `size` as Needed:**
 - 4 Modify the `width`, `height`, and circle `size` to achieve the desired spacing and circle size.
- 5 **Verify Field Mappings:**
 - 6 Ensure `PctTargetSelMTD` is correctly added as a measure in Deneb's Fields pane.
- 7 **Test and Fine-Tune:**
 - 8 Preview the chart and adjust the properties as necessary to get the columns as close as you want without overlapping.

Troubleshooting Tips:

- **Circles Overlapping:**
 - If circles overlap after reducing the chart's width, decrease the circle `size` slightly or increase `paddingInner` a bit.
- **Text Not Centered:**
 - Ensure the `x` value in the text mark's encoding is set to half of the chart's width.

- **Columns Still Spaced Out:**

- Reduce the `width` further or confirm that both `paddingInner` and `paddingOuter` are set to `0` in the x scale.

Final Thoughts:

By setting the chart's `width` and `height`, and adjusting the scales and circle sizes accordingly, you should be able to bring the columns of your waffle chart closer together as desired. Remember that the layout in Vega-Lite is responsive to the chart dimensions, so controlling the size of the chart is key to achieving the exact visual spacing you want.

If you have any further questions or need additional assistance fine-tuning your chart, please don't hesitate to ask. I'm here to help you get your visual exactly as you envision it!

Additional Resources:

- Vega-Lite Documentation on Width and Height:

- Size of View and Marks

- Vega-Lite Documentation on Band Scales:

- Band Scale

- Vega-Lite Documentation on Padding:

- Scale Padding
-

Let me know if these adjustments help resolve the issue!