

Lesson 18

Topic: Visual Calculations

Prerequisites: Download sales_with_geodata.csv file

Puzzle 1: Confusing Totals

In a table with Product, Sales, Quantity, and a calculated column for Sales / Quantity, the total row shows an incorrect value for Sales / Quantity.

Why is the total wrong?

Power BI does not sum up each row's ratio. Instead, it recalculates the measure in the **total row context**, which ignores row-level logic.

Solution: Use SUMX to force row-by-row calculation:

DAX

CopyEdit

Correct Sales per Quantity =

```
DIVIDE(SUM('Sales'[Sales]), SUM('Sales'[Quantity]))
```

Or, if you want the average of each row's sales-per-quantity:

DAX

CopyEdit

Avg Row Sales per Quantity =

```
AVERAGEX(VALUES('Sales'[Product]), DIVIDE([Sales], [Quantity]))
```

Puzzle 2: Filtered vs. Unfiltered Totals

We need to show **two bar chart values**:

- One for **sales per category**
- One ignoring the axis filter (**total sales** for all categories)
- Bonus: % of total

Measure 1: Sales per Category

DAX

CopyEdit

Total Sales = SUM('Sales'[Amount])

Measure 2: All Category Total

DAX

CopyEdit

Total Sales (All Categories) =

CALCULATE([Total Sales], REMOVEFILTERS('Sales'[Category]))

Bonus: % of Total

DAX

CopyEdit

% of Total Sales =

DIVIDE([Total Sales], [Total Sales (All Categories)])

Now the bar chart will show each category's share of overall sales.

Puzzle 3: Changing Context with Slicers

A card visual with Total Sales changes when you select a country from the slicer.

Why does it change?

Because slicers apply filters to the data model. The Total Sales measure respects the filter context — so when you pick a country, only that country's sales are shown.

To create a version that ignores the slicer:

DAX

CopyEdit

Total Sales (All Countries) =

CALCULATE([Total Sales], REMOVEFILTERS('Sales'[Country]))

Now you can place both cards side-by-side: one responds to slicer, one doesn't.

Puzzle 4: Misleading Average

The visual shows Region and Average Sales per Order, but calculating:

DAX

CopyEdit

Average Sales = [Total Sales] / [Total Orders]

gives incorrect results per row.

Why?

Because [Total Sales] and [Total Orders] aggregate over the current row context — but dividing totals doesn't guarantee an accurate **row-level average**.

Fix with AVERAGEX over the orders:

DAX

CopyEdit

Avg Sales per Order =

```
AVERAGEX(
    VALUES('Sales'[OrderID]),
    CALCULATE(SUM('Sales'[Amount]))
)
```

This ensures you're averaging **sales per individual order**, not total sales divided by total orders.

Puzzle 5: Highlight Top Product per Category

You want to show only the **top-selling product** per category in a **Matrix** visual.

Steps:

1. Add Category and Product to Rows
2. Add [Total Sales] to Values
3. Add a visual-level filter on Product
4. Choose **“Top N” → 1**, and base it on [Total Sales]

Now the matrix will only show the **top product for each category**, dynamically.

Puzzle 6: Unexpected Blank Values

Visual: Table showing Customer and a measure for Sales in France

DAX

CopyEdit

```
Sales in France = CALCULATE(SUM(Sales[Sales]), Sales[Country] = "France")
```

Problem: Some customers appear with blank values even though they have purchases.

Reason:

This happens because the customer may not have made **any purchase in France**, and the CALCULATE() filter **removes their rows entirely** (they don't exist under that condition).

Solution: Use the customer context explicitly with a condition:

DAX

CopyEdit

Sales in France =

```
CALCULATE(  
    SUM(Sales[Sales]),  
    FILTER(  
        Sales,  
        Sales[Country] = "France"  
    )  
)
```

This ensures the customer stays in the row context and France-based sales are evaluated within it — fixing the blanks.

Puzzle 7: Time Intelligence Confusion

Visual: Line chart with OrderDate (Month) on the axis

Values: Sales, Previous Month Sales

Goal: Create a line for previous month's sales and handle missing months.

Solution Measure:

DAX

CopyEdit

Previous Month Sales =

```
CALCULATE(  
    SUM(Sales[Sales]),  
    PARALLELPERIOD('Calendar'[Date], -1, MONTH)  
)
```

Notes:

- Make sure to use a **proper Date table** ('Calendar') marked as a Date Table.
 - PARALLELPERIOD() handles year transitions (e.g., Dec → Jan) better than shifting months manually.
 - If months are missing, use a complete calendar and fill gaps with 0 for better continuity.
-

Puzzle 8: Row-Level Calculation

Visual: Table with Product, Quantity, Discount per Unit, Total Discount

Why use SUMX() instead of a basic multiplication?

DAX

CopyEdit

Total Discount = SUMX(Sales, Sales[Quantity] * Sales[Discount per Unit])

Explanation:

SUMX() is needed when you want to **multiply per row and then add**, which is different from:

DAX

CopyEdit

SUM(Sales[Quantity]) * SUM(Sales[Discount per Unit]) ❌ (Incorrect)

That multiplies totals, not row-level values, which gives **wrong results** unless all rows are identical.

Puzzle 9: Rank with Ties

Visual: Table showing City, Total Sales, and Rank

Measure to handle ties and direction:

DAX

CopyEdit

Sales Rank =

RANKX(

ALL('Sales'[City]),

```
CALCULATE(SUM(Sales[Sales])),  
  
,  
  
DESC, -- Or ASC for ascending order  
  
DENSE  
)
```

Explanation:

- DENSE keeps the same rank for tied values and avoids skipping numbers.
- You can switch between ASC or DESC to control ranking direction.

Puzzle 10: Dynamic Titles and KPIs

Visuals: Card with dynamic title responding to slicer (e.g., selected country)

Measure for dynamic title:

DAX

CopyEdit

Dynamic Title =

"Sales for " & SELECTEDVALUE(Sales[Country], "All Countries")

Now the card title will say:

- "Sales for USA" when a country is selected
- "Sales for All Countries" when nothing is selected