

# Visual Calculations — Theoretical Answers & DAX Patterns

Dataset: sales\_with\_geodata.csv

This document explains why certain visuals behave unexpectedly in Power BI and provides robust DAX patterns to get correct totals and context-aware results. All measures assume a fact table named **Sales** with columns like **Sales**, **Quantity**, **OrderDate**, **Country**, **Category**, **Product**, **Customer**, **Region**, **City**, **Discount per Unit**, etc. Create a proper Calendar/Date dimension (e.g., table 'Date') and relate it to **Sales[OrderDate]** for time-intelligence.

## Puzzle 1: Confusing Totals (Sales / Quantity)

**Why totals differ:** A per-row division ( $\text{Sales} \div \text{Quantity}$ ) summed across rows is not the same as dividing grand totals. In the total row, the filter context is the whole table, so a measure like *`DIVIDE(SUM(Sales[Sales]), SUM(Sales[Quantity]))`* returns a ratio of sums, which rarely equals the sum of row-level ratios shown above.

### Robust pattern 1 — Ratio of sums (standard):

```
Total Sales := SUM(Sales[Sales])
Total Qty   := SUM(Sales[Quantity])
Sales per Qty (ratio of sums) := DIVIDE([Total Sales], [Total Qty])
```

### Robust pattern 2 — Sum of row-level ratios (matches visible rows):

```
Sales per Qty (sum of ratios) :=
SUMX(
    VALUES(Sales[Product]),
    DIVIDE([Total Sales], [Total Qty])
)
```

Pick the pattern that matches your business definition.

## Puzzle 2: Filtered vs. Unfiltered Totals (Bar by Category)

**Goal:** Show sales per category and also a line/column that ignores the Category axis filter to always show the grand total. Then compute % of total.

```
Total Sales := SUM(Sales[Sales])
```

```
Total Sales (All Categories) :=
CALCULATE(
    [Total Sales],
    ALL(Sales[Category])
)
```

```
% of Total (by Category) :=
DIVIDE([Total Sales], [Total Sales (All Categories)])
```

## Puzzle 3: Changing Context with Slicers (Card + Country slicer)

**Why the card changes:** Slicers modify filter context. Selecting a country filters the measure to that country, so the card updates. **To ignore the slicer:** Remove the Country filter inside the measure.

```
Total Sales := SUM(Sales[Sales])
```

```
Total Sales (Ignore Country) :=
CALCULATE(
    [Total Sales],
```

```

    REMOVEFILTERS(Sales[Country])
)

```

## Puzzle 4: Misleading Average (Region table)

**Problem:** Using  $[Total\ Sales] / [Total\ Orders]$  divides two aggregates at the current context (ratio of sums). In a visual, this can produce a different total than the simple average of per-order amounts and can be misleading across groups. **Fix:** Compute the mean at the correct grain with AVERAGEX over orders (or any desired granularity), then aggregate.

```
Total Orders := DISTINCTCOUNT(Sales[OrderID])
```

```

Avg Sales per Order (correct) :=
AVERAGEX(
    VALUES(Sales[OrderID]),
    CALCULATE([Total Sales])
)

```

-- Optional: if you specifically want ratio-of-sums in each context:

```
Avg Sales per Order (ratio-of-sums) := DIVIDE([Total Sales], [Total Orders])
```

## Puzzle 5: Highlight Top Product per Category (Matrix)

**Goal:** Show only the top-selling product within each category. Visual-level Top N can't partition by category automatically, so use a ranking measure partitioned by Category and filter Rank = 1.

```
Total Sales := SUM(Sales[Sales])
```

```

Rank Product in Category :=
VAR CurCat = SELECTEDVALUE(Sales[Category])
RETURN
RANKX(
    FILTER(
        ALLSELECTED(Sales[Product], Sales[Category]),
        Sales[Category] = CurCat
    ),
    [Total Sales],
    ,
    DESC,
    DENSE
)

```

-- In the Matrix's visual-level filters: set [Rank Product in Category] = 1.

## Puzzle 6: Unexpected Blank Values (Sales in France)

**Why blanks appear:** The measure restricts to Country = "France". Customers who have no French purchases will be blank (not zero). Existing page/visual filters may also conflict with the Country filter.

**Fix:** Remove the current country filter first, then apply France; return 0 instead of blank if preferred.

```

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

```

```

Sales in France (0 if none) :=
COALESCE([Sales in France], 0)

```

## Puzzle 7: Time Intelligence Confusion (Previous Month Sales)

**Requirement:** Plot monthly Sales and a prior-month line. **Important:** Use a contiguous 'Date' table related to Sales[OrderDate]. DATEADD shifts context; it returns blank at the earliest month (no previous). Handle that gracefully.

```
-- Prereq: 'Date'[Date] is related to Sales[OrderDate]
Total Sales := SUM(Sales[Sales])
```

```
Previous Month Sales :=
CALCULATE(
    [Total Sales],
    DATEADD('Date'[Date], -1, MONTH)
)
```

```
Previous Month Sales (0 if first month) :=
COALESCE([Previous Month Sales], 0)
```

If months are missing in data, the 'Date' table ensures a complete calendar so the line chart remains continuous.

## Puzzle 8: Row-Level Calculation (Total Discount)

**Why SUMX():** Discounts are row-level: you must multiply Quantity × Discount per Unit per row, then sum. Multiplying aggregates like SUM(Quantity) × SUM(Discount per Unit) is mathematically wrong unless discount per unit is constant.

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

## Puzzle 9: Rank with Ties (City)

**Goal:** Rank cities by Total Sales with proper tie handling. Use ALLSELECTED to respect user filters; choose DENSE or SKIP ties as needed.

```
Total Sales := SUM(Sales[Sales])
```

```
Rank (City by Sales, Desc, Dense) :=
RANKX(
    ALLSELECTED(Sales[City]),
    [Total Sales],
    ,
    DESC,
    DENSE
)
```

```
-- For ascending order:
Rank (City by Sales, Asc, Dense) :=
RANKX(
    ALLSELECTED(Sales[City]),
    [Total Sales],
    ,
    ASC,
    DENSE
)
```

## Puzzle 10: Dynamic Titles and KPIs

**Goal:** Display a dynamic title that reflects the selected country; show "All Countries" when none or multiple are selected.

Title (by Country) :=

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

**General Tips**

- Prefer explicit measures over implicit aggregations.
- Use REMOVEFILTERS/ALL to control when visuals should ignore slicers/axes.
- For “averages”, validate the intended grain (per row, per order, per customer) and use AVERAGEX or SUMX as needed.
- Always create a proper 'Date' table for time intelligence.
- For partitioned Top N (e.g., per Category), use a rank measure with FILTER on the partition key.