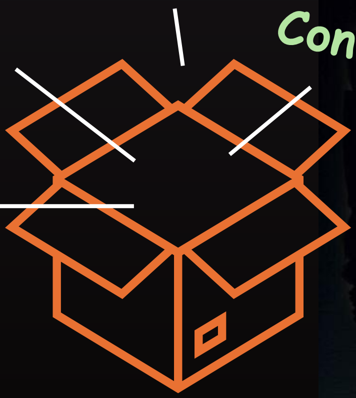
Start with PowerBI

"From Rookie to Rock"

Learn &
Practice

Quick Guide of CALCULATE



Patou Tips #48

CALCULATE
Summary

If you only need to know and understand one DAX formula, make it this one. **CALCULATE** is the most powerful function in the DAX language. Here a Quick Guide to explore step by step how this function works.

- ✓ Part 1 - Substitution role, Calculation with “FILTER”
- ✓ Part 2 - Context transition, Calculation with ALL, ALLEXCEPT, ALLSELECTED
- ✓ Part 3 - Context transition & Iteration
- ✓ Part 4 - Rules Summary
- ✓ Part 5 - Useful Patterns

Quick Guide of CALCULATE

Part 1

Substitution role, Calculation with “FILTER”

Start with PowerBI

“From Rookie to Rock”

Patou Tips #48

About CALCULATE (1/3)
Syntax and definition

If you only need to know and understand one DAX formula, make it this one. CALCULATE is the most powerful function in the DAX language. In this new series we will explore step by step how this function works.

Syntax

```
CALCULATE ( Expression, Filter 1, Filter 2, …, Filter n)
```

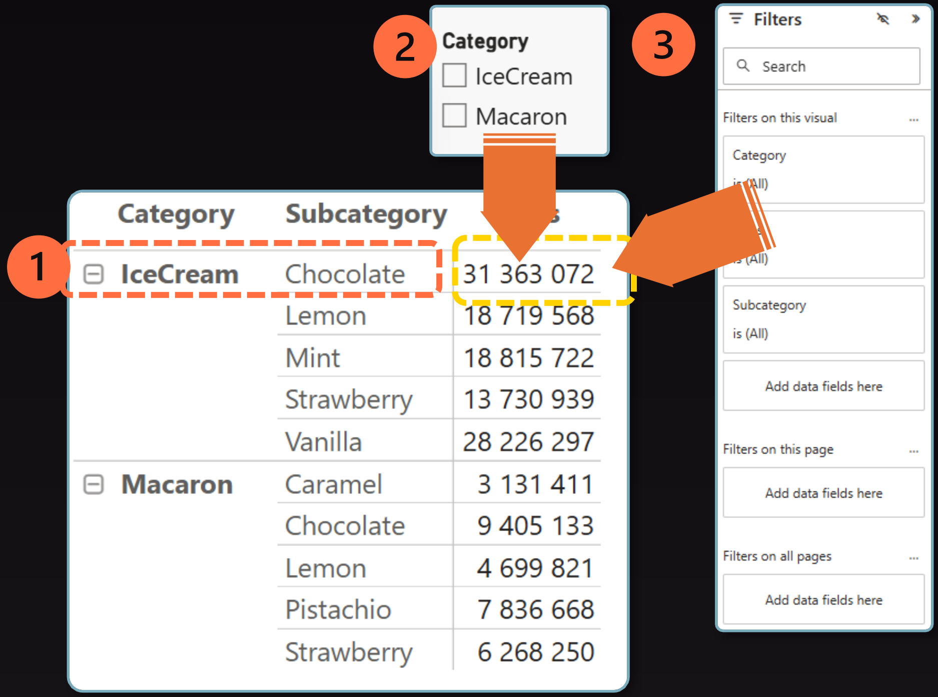
Definition

Evaluates an expression based on a context modified by filters and returns a single value of any type (Scalar).

- ✓ Note 1: The project's modeling structure must be taken into account.
- ✓ Note 2: The number of filters applied can be very large.

About CALCULATE (2/3)
Row and filter context notion

The CALCULATE function is like a radar that takes into account the elements (filters) available in its environment...



Category
☐ IceCream
☐ Macaron

Category	Subcategory	[illegible]
⊟ IceCream	Chocolate	31 363 072
	Lemon	18 719 568
	Mint	18 815 722
	Strawberry	13 730 939
	Vanilla	28 226 297
⊟ Macaron	Caramel	3 131 411
	Chocolate	9 405 133
	Lemon	4 699 821
	Pistachio	7 836 668
	Strawberry	6 268 250

Filters

Search

Filters on this visual ...

Category
is (All)

is (All)

Subcategory
is (All)

Add data fields here

Filters on this page ...

Add data fields here

Filters on all pages ...

Add data fields here

The value

31 363 072

is the result of

Row context

1

(IceCream and Chocolate)

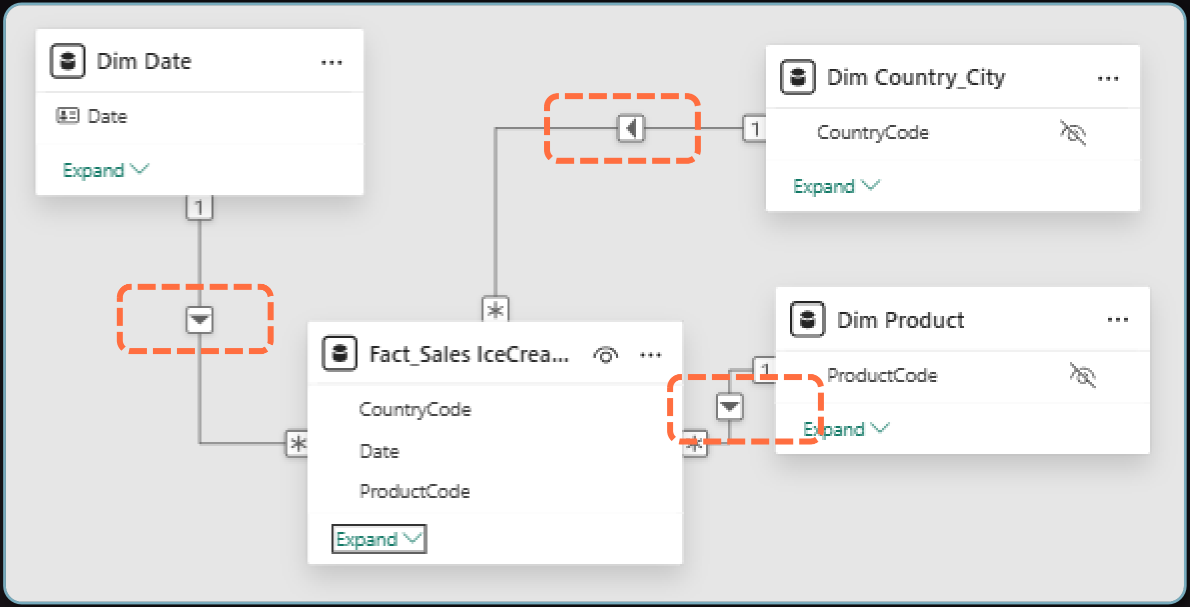
Filter context

2 3

Slicer or Filters pane

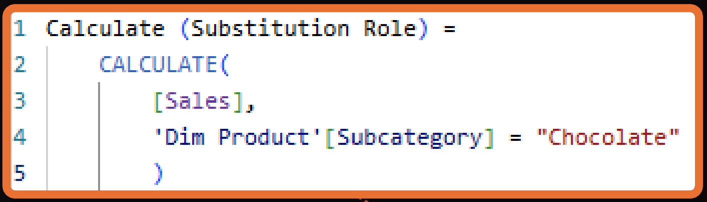
About CALCULATE (2/3)
Structure of the modeling

The CALCULATE function also depends on the project modeling. The "Calculate" function will activate data flows between dimension tables and fact tables.

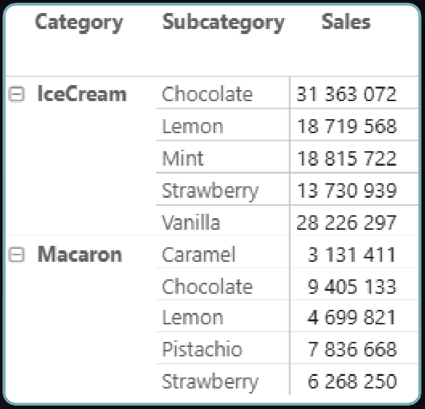


CALCULATE
Substitution role (1/2)

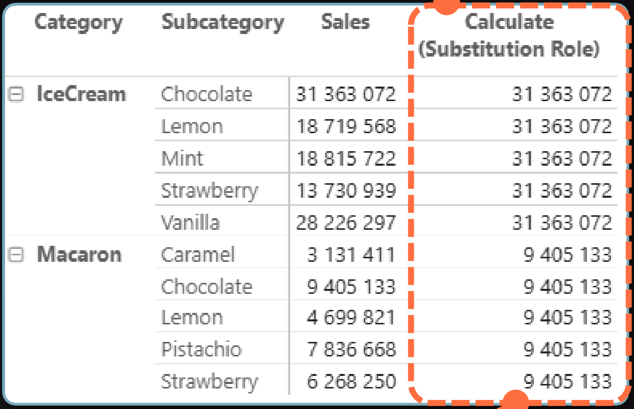
This formula is really interesting, to create a substitute role to better understand the context line.



```
Calculate (Substitution Role) =
    CALCULATE(
        [Sales],
        'Dim Product'[Subcategory] = "Chocolate"
        )
```



Category	Subcategory	Sales
⊟ IceCream	Chocolate	31 363 072
	Lemon	18 719 568
	Mint	18 815 722
	Strawberry	13 730 939
	Vanilla	28 226 297
⊟ Macaron	Caramel	3 131 411
	Chocolate	9 405 133
	Lemon	4 699 821
	Pistachio	7 836 668
	Strawberry	6 268 250



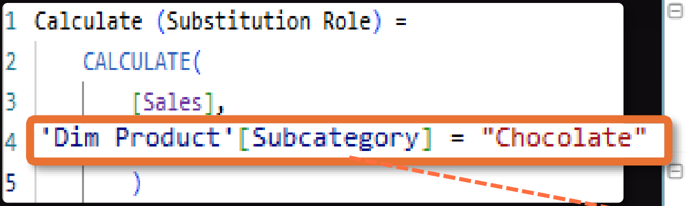
Category	Subcategory	Sales	Calculate (Substitution Role)
⊟ IceCream	Chocolate	31 363 072	31 363 072
	Lemon	18 719 568	31 363 072
	Mint	18 815 722	31 363 072
	Strawberry	13 730 939	31 363 072
	Vanilla	28 226 297	31 363 072
⊟ Macaron	Caramel	3 131 411	9 405 133
	Chocolate	9 405 133	9 405 133
	Lemon	4 699 821	9 405 133
	Pistachio	7 836 668	9 405 133
	Strawberry	6 268 250	9 405 133

?

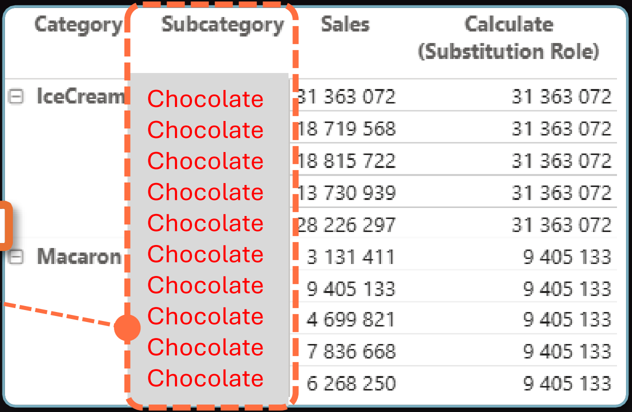
The expected result displays values on all rows, not just the "Chocolate" rows.

CALCULATE
Substitution role (2/2)

Explanations about the behavior of the CALCULATE function



```
Calculate (Substitution Role) =
    CALCULATE(
        [Sales],
 'Dim Product'[Subcategory] = "Chocolate"
        )
```

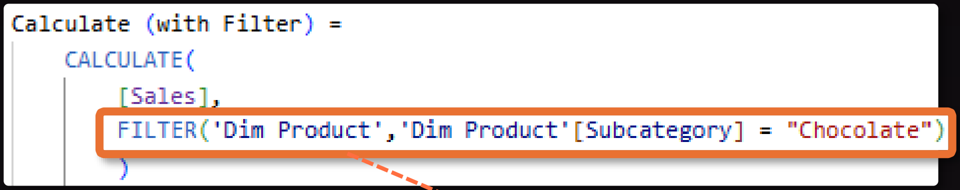


Category	Subcategory	Sales	Calculate (Substitution Role)
⊟ IceCream	Chocolate	31 363 072	31 363 072
	Chocolate	18 719 568	31 363 072
	Chocolate	18 815 722	31 363 072
	Chocolate	13 730 939	31 363 072
	Chocolate	28 226 297	31 363 072
⊟ Macaron	Chocolate	3 131 411	9 405 133
	Chocolate	9 405 133	9 405 133
	Chocolate	4 699 821	9 405 133
	Chocolate	7 836 668	9 405 133
	Chocolate	6 268 250	9 405 133

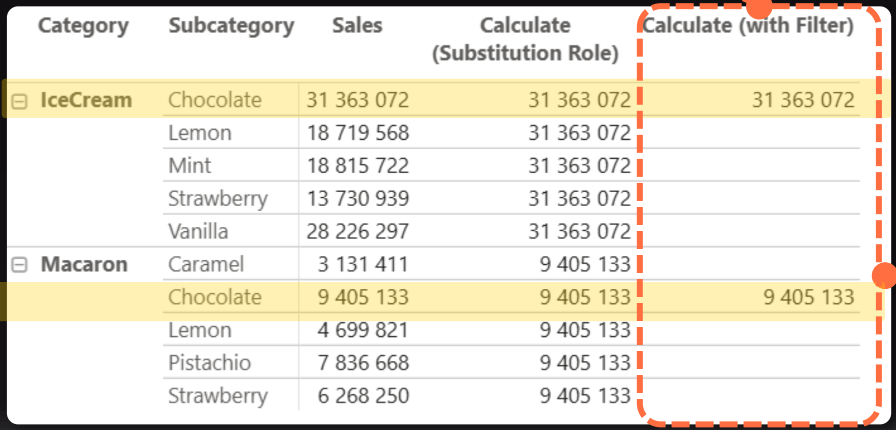
→ For each row, "Calculate" replaces the value in the "Subcategory" column with the value "Chocolate."
→ For each row, the result depends on the value in the "Category" field (here, Ice Cream or Macaron), and for the "Subcategory" value, "Calculate" replaces the value with the value "Chocolate."
→ This is why the result corresponds to the value of Chocolate in each "Category."

CALCULATE
Calculation with “FILTER”

To create a formula that allows you to filter the previous result only on the "Chocolate" lines, the combination with the "FILTER" function is perfect.



```
Calculate (with Filter) =
    CALCULATE(
        [Sales],
        FILTER('Dim Product','Dim Product'[Subcategory] = "Chocolate")
        )
```



Category	Subcategory	Sales	Calculate (Substitution Role)	Calculate (with Filter)
⊟ IceCream	Chocolate	31 363 072	31 363 072	31 363 072
	Lemon	18 719 568	31 363 072	
	Mint	18 815 722	31 363 072	
	Strawberry	13 730 939	31 363 072	
	Vanilla	28 226 297	31 363 072	
⊟ Macaron	Caramel	3 131 411	9 405 133	
	Chocolate	9 405 133	9 405 133	9 405 133
	Lemon	4 699 821	9 405 133	
	Pistachio	7 836 668	9 405 133	
	Strawberry	6 268 250	9 405 133	

The expected result displays values only in the "Chocolate" rows.

Quick Guide of CALCULATE

Part 2

Context transition, Calculation with ALL, ALLEXCEPT, ALLSELECTED

Start with PowerBI

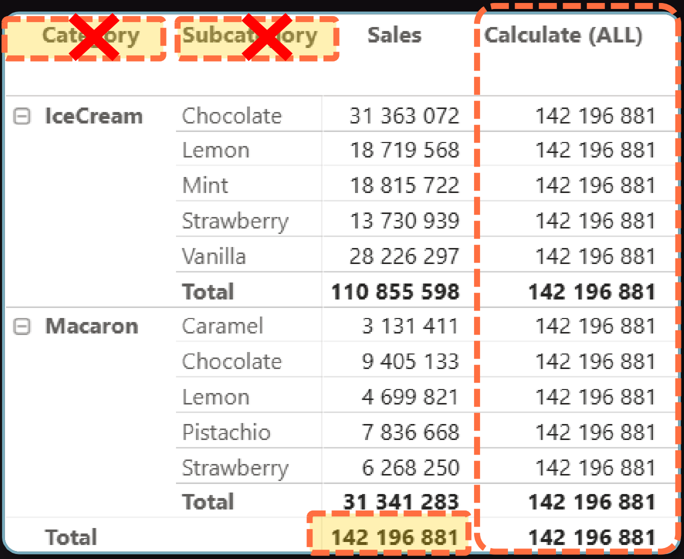
"From Rookie to Rock"

Patou Tips #48

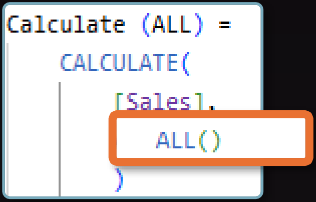
CALCULATE
Calculation with ALL (1/2)

ALL remove all the filter specified inside the function. When nothing is specified, ALL remove all the filters.

Only ALL



Category	Subcategory	Sales	Calculate (ALL)
⊟ IceCream	Chocolate	31 363 072	142 196 881
	Lemon	18 719 568	142 196 881
	Mint	18 815 722	142 196 881
	Strawberry	13 730 939	142 196 881
	Vanilla	28 226 297	142 196 881
	Total	**110 855 598**	**142 196 881**
⊟ Macaron	Caramel	3 131 411	142 196 881
	Chocolate	9 405 133	142 196 881
	Lemon	4 699 821	142 196 881
	Pistachio	7 836 668	142 196 881
	Strawberry	6 268 250	142 196 881
	Total	**31 341 283**	**142 196 881**
Total		**142 196 881**	**142 196 881**



```
Calculate (ALL) =
    CALCULATE(
        [Sales],
        ALL()
    )
```

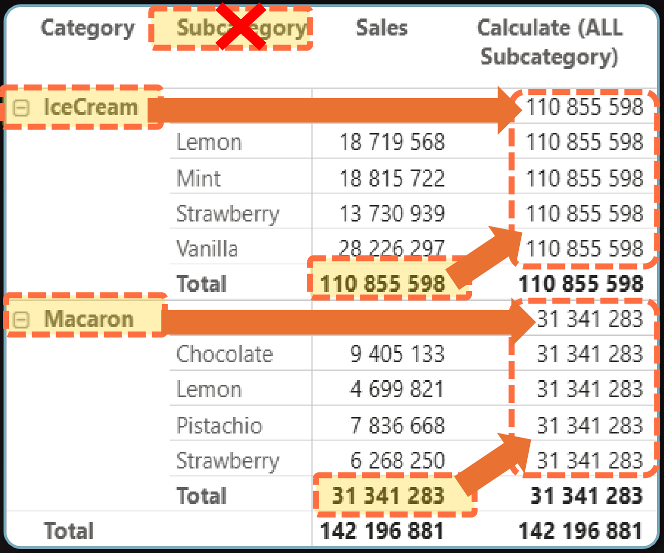
If nothing is specified, ALL removes all filters. The total "Sales" is finally displayed for all rows.

✏️ Note: ALL et REMOVEFILTERS are the same function.

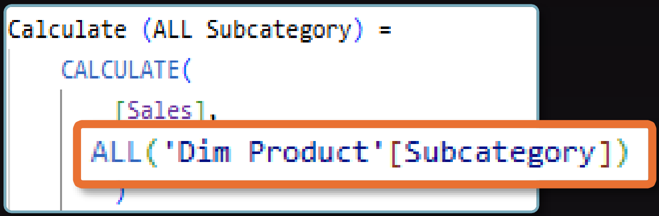
CALCULATE
Context transition with ALL

In this example with an argument inside ALL, we begin to get a better feel for the context transition process.

ALL with argument



Category	Subcategory	Sales	Calculate (ALL Subcategory)
IceCream			110 855 598
	Lemon	18 719 568	110 855 598
	Mint	18 815 722	110 855 598
	Strawberry	13 730 939	110 855 598
	Vanilla	28 226 297	110 855 598
	Total	**110 855 598**	**110 855 598**
Macaron			31 341 283
	Chocolate	9 405 133	31 341 283
	Lemon	4 699 821	31 341 283
	Pistachio	7 836 668	31 341 283
	Strawberry	6 268 250	31 341 283
	Total	**31 341 283**	**31 341 283**
Total		**142 196 881**	**142 196 881**



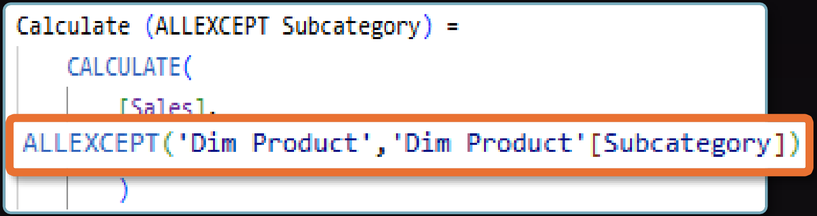
```
Calculate (ALL Subcategory) =
    CALCULATE(
        [Sales],
        ALL('Dim Product'[Subcategory])
    )
```

Only the "Subcategory" filter is ignored. Therefore, for each "Subcategory," the "Sales" total is displayed for all rows.

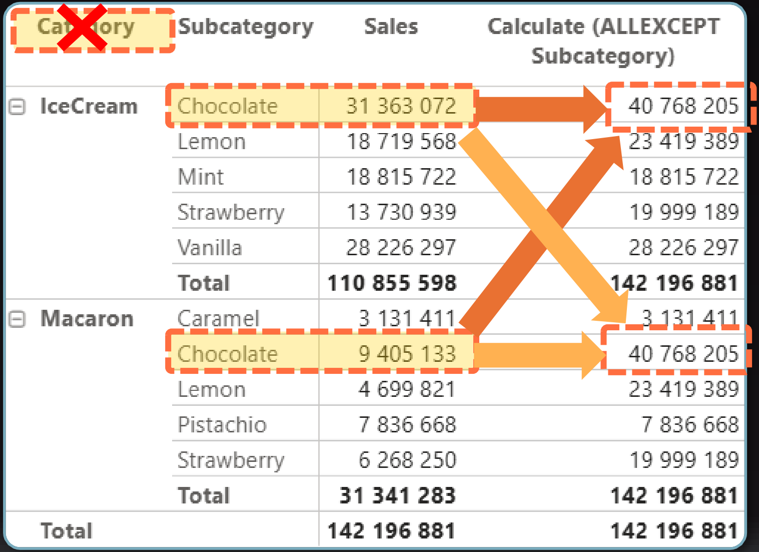
The "Category" filter is only considered for each row; this is the context transition.

CALCULATE
Context transition with ALLEXCEPT

Context transition is important to understand how the calculations will work. Let's go further with another example.



```
Calculate (ALLEXCEPT Subcategory) =
    CALCULATE(
        [Sales],
ALLEXCEPT('Dim Product','Dim Product'[Subcategory])
        )
```



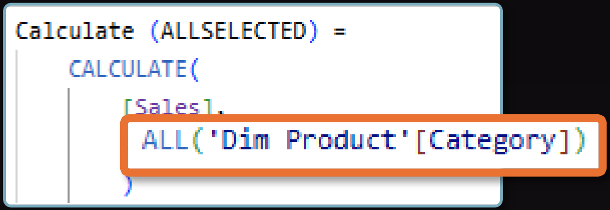
Category	Subcategory	Sales	Calculate (ALLEXCEPT Subcategory)
⊟ IceCream	Chocolate	31 363 072	40 768 205
	Lemon	18 719 568	23 419 389
	Mint	18 815 722	18 815 722
	Strawberry	13 730 939	19 999 189
	Vanilla	28 226 297	28 226 297
	Total	**110 855 598**	**142 196 881**
⊟ Macaron	Caramel	3 131 411	3 131 411
	Chocolate	9 405 133	40 768 205
	Lemon	4 699 821	23 419 389
	Pistachio	7 836 668	7 836 668
	Strawberry	6 268 250	19 999 189
	Total	**31 341 283**	**142 196 881**
Total		**142 196 881**	**142 196 881**

Only the "Subcategory" filter is taken into account. Therefore, for each "Subcategory" value, the context transition on the sales value will be applied by the same "Subcategory" value.

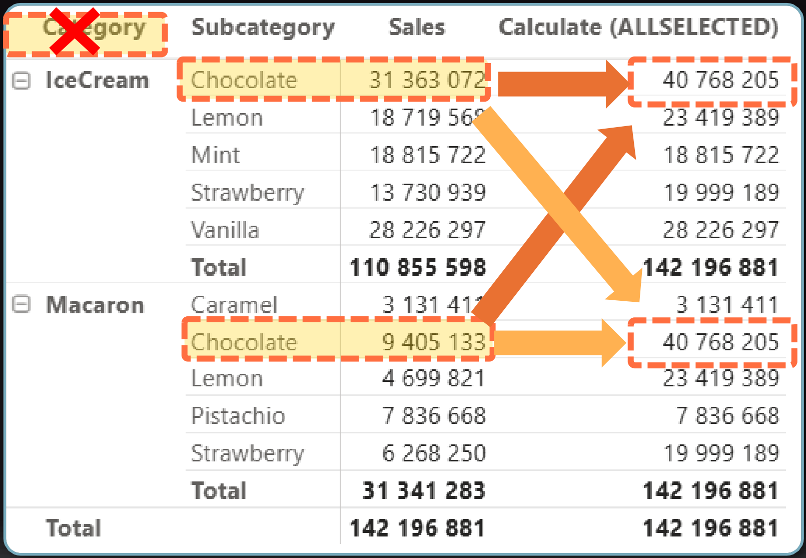
For the Chocolate "Subcategory", the total corresponds to the sum of all Chocolate values in the list. The process is identical for Lemon.

CALCULATE
Context transition with ALLSELECTED

We get the same behavior, context transition with ALL, the explanation is exactly the same as before.



```
Calculate (ALLSELECTED) =
    CALCULATE(
        [Sales],
        ALL('Dim Product'[Category])
    )
```



Category	Subcategory	Sales	Calculate (ALLSELECTED)
IceCream	Chocolate	31 363 072	40 768 205
	Lemon	18 719 568	23 419 389
	Mint	18 815 722	18 815 722
	Strawberry	13 730 939	19 999 189
	Vanilla	28 226 297	28 226 297
	Total	**110 855 598**	**142 196 881**
Macaron	Caramel	3 131 411	3 131 411
	Chocolate	9 405 133	40 768 205
	Lemon	4 699 821	23 419 389
	Pistachio	7 836 668	7 836 668
	Strawberry	6 268 250	19 999 189
	Total	**31 341 283**	**142 196 881**
Total		**142 196 881**	**142 196 881**

Only the "Subcategory" filter is taken into account. Therefore, for each "Subcategory" value, the context transition on the sales value will be applied by the same "Subcategory" value.

For the Chocolate "Subcategory", the total corresponds to the sum of all Chocolate values in the list. The process is identical for Lemon.

Quick Guide of CALCULATE

Part 3

Context transition & Iteration

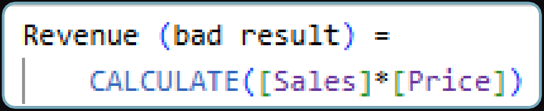
Start with PowerBI

"From Rookie to Rock"

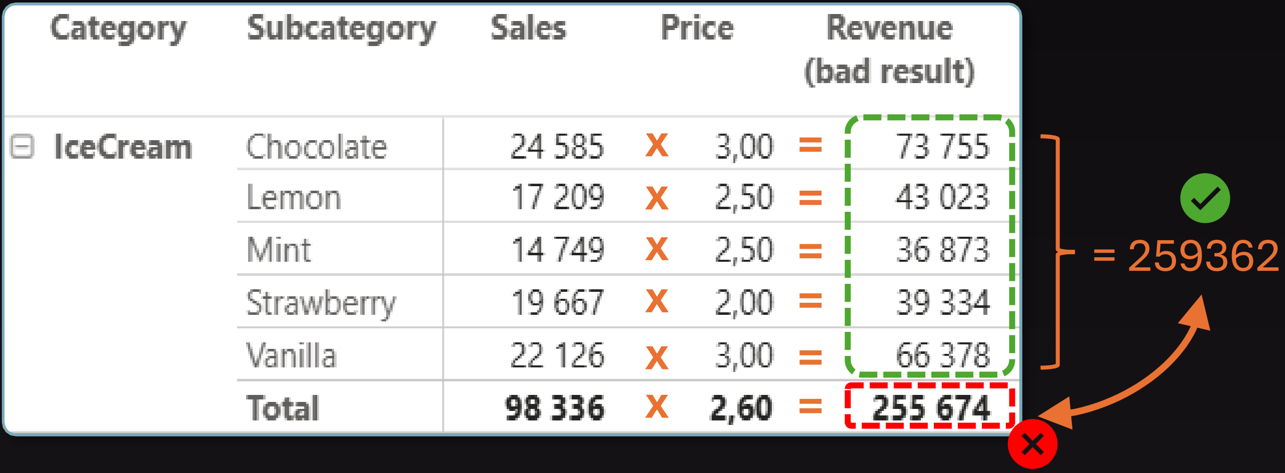
Patou Tips #48

CALCULATE
Context transition & Iteration

When you start to go deeper into CALCULATE, you quickly run into the problem of false results. The problem comes from often of a transition context without reiteration.



```
Revenue (bad result) =
    CALCULATE([Sales]*[Price])
```



Category	Subcategory	Sales		Price		Revenue (bad result)
⊟ IceCream	Chocolate	24 585	X	3,00	=	73 755
	Lemon	17 209	X	2,50	=	43 023
	Mint	14 749	X	2,50	=	36 873
	Strawberry	19 667	X	2,00	=	39 334
	Vanilla	22 126	X	3,00	=	66 378
	Total	**98 336**	X	**2,60**	=	**255 674**

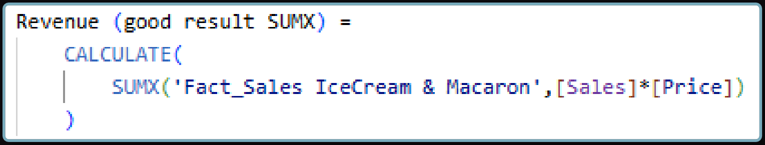
✓ = 259362

✗

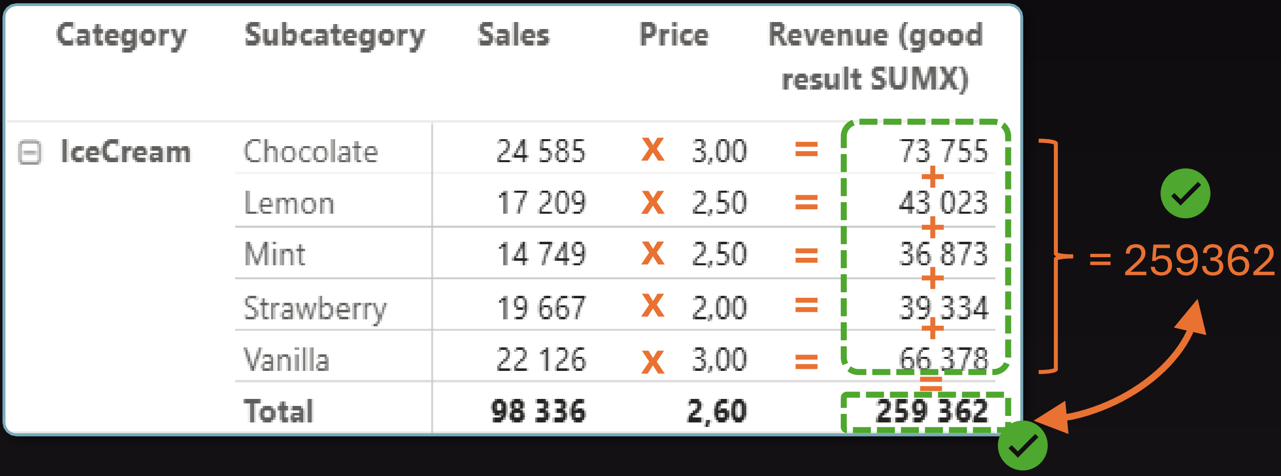
Each row gives a good result, but the problem comes from the total where the same rule is applied and ultimately gives a bad result.

CALCULATE
Context transition & Iteration

Rule → Context Transition is triggered by a Row Context.



```
Revenue (good result SUMX) =
    CALCULATE(
    |   SUMX('Fact_Sales IceCream & Macaron',[Sales]*[Price])
    )
```



Category	Subcategory	Sales		Price		Revenue (good result SUMX)
⊟ IceCream	Chocolate	24 585	X	3,00	=	73 755
	Lemon	17 209	X	2,50	=	43 023
	Mint	14 749	X	2,50	=	36 873
	Strawberry	19 667	X	2,00	=	39 334
	Vanilla	22 126	X	3,00	=	66 378
	Total	**98 336**		**2,60**		**259 362**

To get a good result, combine CALCULATE with an iteration function such as SUMX.

Explanation: For each row, CALCULATE with SUMX applies the Row Context ([Sales] x [Price]) to the table (Fact_Sales IceCream & Macaron) and iterates through each row until the final total is reached.

Start with PowerBI
"From Rookie to Rock"

Part 4

Rules Summary

CALCULATE
Evaluation Context

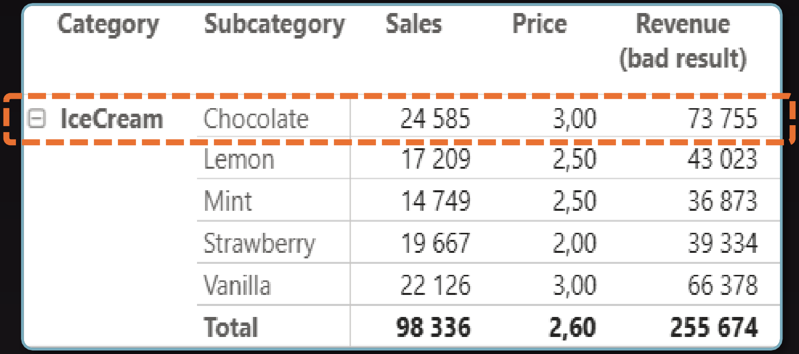
Evaluation Context

The **Evaluation Context** is composed of:

- ✓ Row context
- ✓ Filter context:

Row Context

Determines the row where the calculation is performed



Category	Subcategory	Sales	Price	Revenue (bad result)
⊟ **IceCream**	Chocolate	24 585	3,00	73 755
	Lemon	17 209	2,50	43 023
	Mint	14 749	2,50	36 873
	Strawberry	19 667	2,00	39 334
	Vanilla	22 126	3,00	66 378
	Total	**98 336**	**2,60**	**255 674**

Iteration functions (SUMX, AVERAGEX,...) read each line to create a line context which will trigger the transition context and iteration at each line.

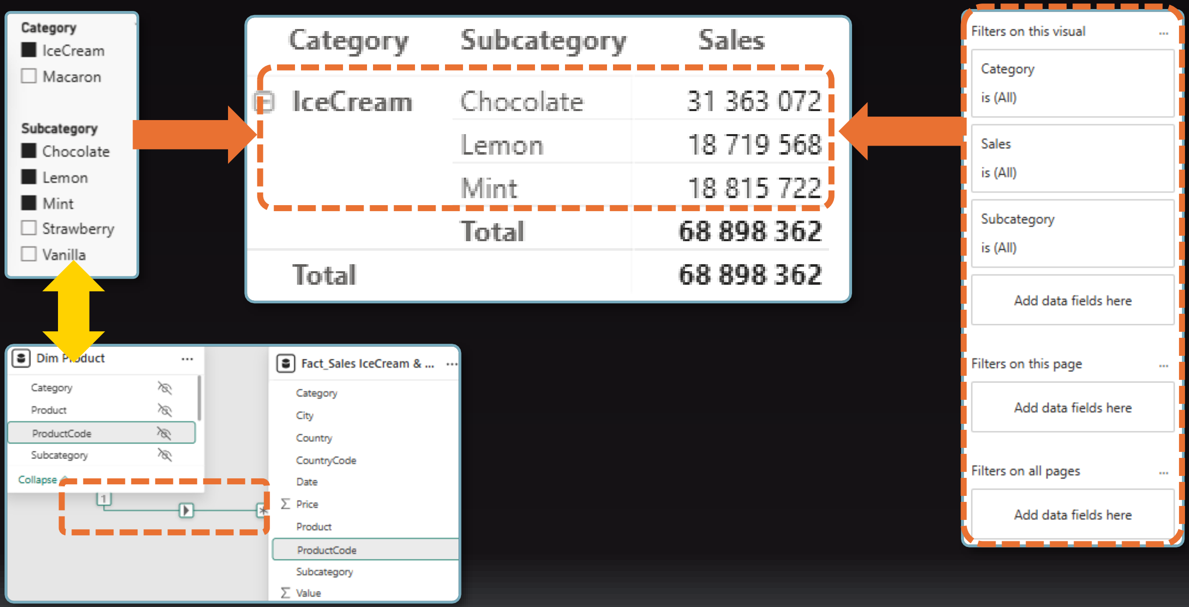
CALCULATE
Evaluation Context

Filter Context

Determines the visible rows and the scope of the calculation

Filtering contexts can be:

- ✓ Direct: via visual elements, filtering options such as slicers, filter panels, or modeling
- ✓ Indirect: via filter functions such as ALL (REMOVEFILTER), FILTER, ALLEXCEPT, etc.



Category
- ■ IceCream
- □ Macaron

Subcategory
- ■ Chocolate
- ■ Lemon
- ■ Mint
- □ Strawberry
- □ Vanilla

Category	Subcategory	Sales
IceCream	Chocolate	31 363 072
	Lemon	18 719 568
	Mint	18 815 722
	Total	68 898 362
Total		68 898 362

Filters on this visual

Category
is (All)

Sales
is (All)

Subcategory
is (All)

Add data fields here

Filters on this page

Add data fields here

Filters on all pages

Add data fields here

Dim Product
- Category
- Product
- ProductCode
- Subcategory

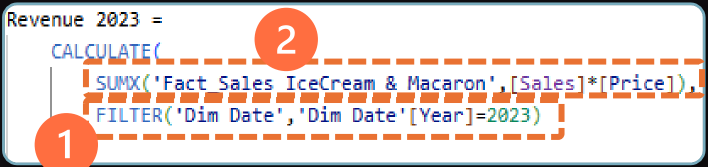
Collapse

Fact_Sales IceCream & ...
- Category
- City
- Country
- CountryCode
- Date
- Σ Price
- Product
- ProductCode
- Subcategory
- Σ Value

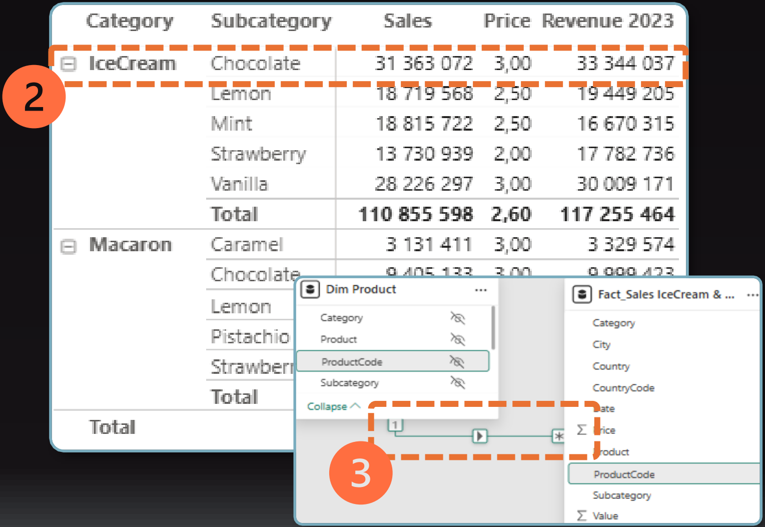
CALCULATE
Context transition

Context Transition

Context transition is the calculation process but also the one that activates modeling relationships.



```
Revenue 2023 =
    CALCULATE(
        SUMX('Fact_Sales IceCream & Macaron',[Sales]*[Price]),
        FILTER('Dim Date','Dim Date'[Year]=2023)
```



Category	Subcategory	Sales	Price	Revenue 2023
⊟ IceCream	Chocolate	31 363 072	3,00	33 344 037
	Lemon	18 719 568	2,50	19 449 205
	Mint	18 815 722	2,50	16 670 315
	Strawberry	13 730 939	2,00	17 782 736
	Vanilla	28 226 297	3,00	30 009 171
	Total	**110 855 598**	**2,60**	**117 255 464**
⊟ Macaron	Caramel	3 131 411	3,00	3 329 574
	Chocolate	9 405 133	3,00	9 999 423
	Lemon			
	Pistachio			
	Strawberr			
	Total			
Total				

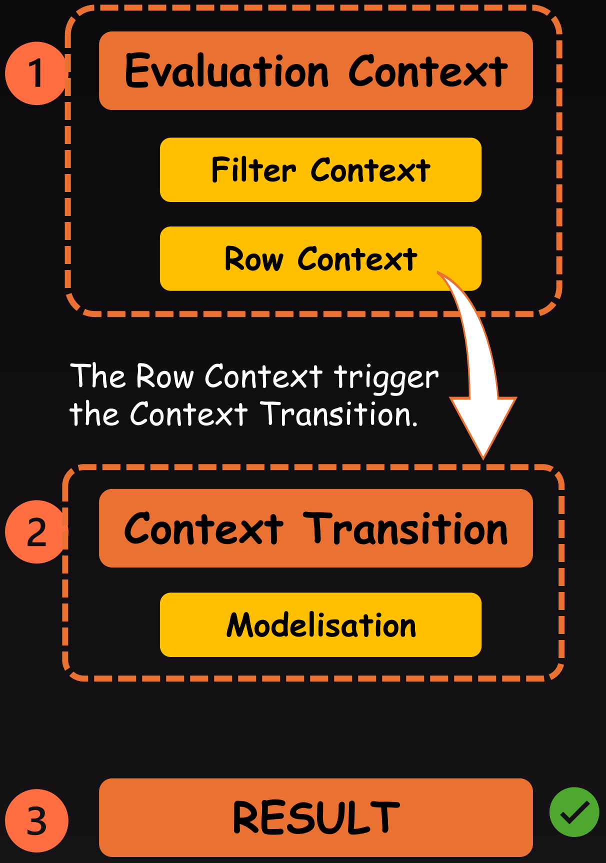
1. How does it work? The calculation takes into account the Context Filter for the Evaluation Context, here the year "2023".

2. Then, still for the Evaluation Context, it takes into account the iteration function applied to each row. This Row Context triggers the Context Transition.

3. The Context Transition calculates the formula and activates all relationships according to the modeling.

CALCULATE
Rules (synthesis)

The Evaluation Context (Filter + Row Context) determines the arguments to be taken into account and the calculation scope. The Row Context then will trigger the context transition.



1 Evaluation Context

Filter Context

Row Context

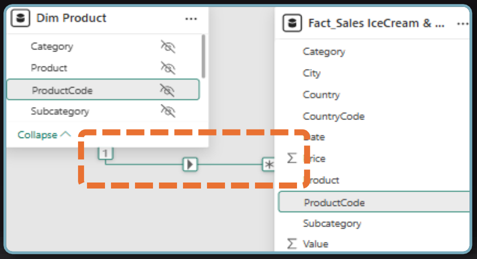
The Row Context trigger the Context Transition.

2 Context Transition

Modelisation

3 RESULT ✓

When the Transition Context is triggered, the modeling relationship will apply: cardinality, cross-filter direction.



Quick Guide of CALCULATE

Part 5

Useful Patterns

Start with PowerBI

"From Rookie to Rock"

Patou Tips #48

CALCULATE
Useful Patterns

In this new chapter on the DAX "Calculate" function, we will look at some very useful patterns, applying the rules relating to filter and row contexts, and context transition, which we have previously discussed.

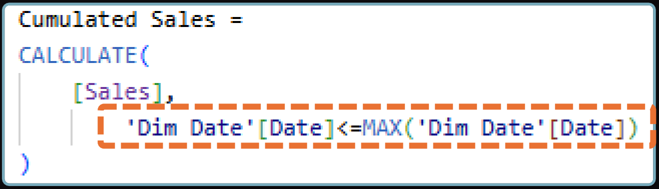
- ✓ Cumulated total (in Time Series)
- ✓ Cumulated total (with Product dimension)
- ✓ Average (in Time Series)
- ✓ Average (with Product dimension)
- ✓ Patterns with Time Intelligence
- ✓ Patterns for Min, Max, First, Last values
- ✓ Use Treatas to connect 2 differents data sources

CALCULATE

Cumulated total (in Time Series)

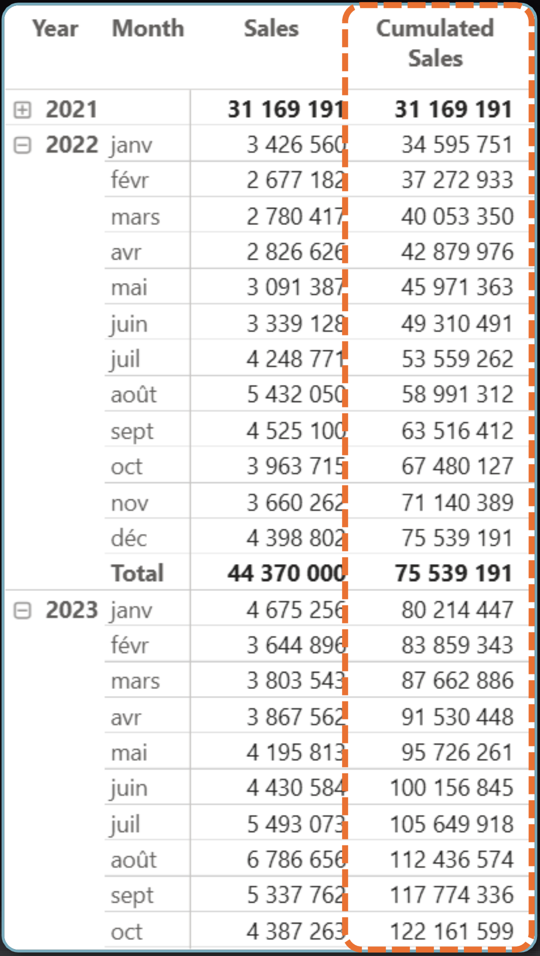
Cumulated Sales

This pattern is interesting to build a cumulated total



```
Cumulated Sales =
CALCULATE(
    [Sales],
    'Dim Date'[Date]<=MAX('Dim Date'[Date])
)
```

In this pattern, the filter is a loop to create a row context. This loop compares the date of the row with the maximum of the date of the selection (the table) for each row. This row context triggers the context transition, so for each row the condition is true, so it performs the cumulative sales for each row.



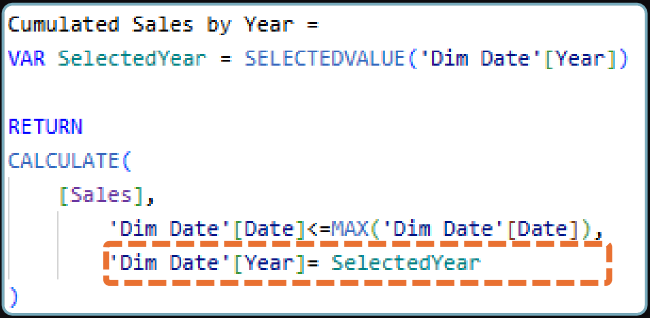
Year	Month	Sales	Cumulated Sales
⊞ **2021**		**31 169 191**	**31 169 191**
⊟ **2022**	janv	3 426 560	34 595 751
	févr	2 677 182	37 272 933
	mars	2 780 417	40 053 350
	avr	2 826 626	42 879 976
	mai	3 091 387	45 971 363
	juin	3 339 128	49 310 491
	juil	4 248 771	53 559 262
	août	5 432 050	58 991 312
	sept	4 525 100	63 516 412
	oct	3 963 715	67 480 127
	nov	3 660 262	71 140 389
	déc	4 398 802	75 539 191
	Total	**44 370 000**	**75 539 191**
⊟ **2023**	janv	4 675 256	80 214 447
	févr	3 644 896	83 859 343
	mars	3 803 543	87 662 886
	avr	3 867 562	91 530 448
	mai	4 195 813	95 726 261
	juin	4 430 584	100 156 845
	juil	5 493 073	105 649 918
	août	6 786 656	112 436 574
	sept	5 337 762	117 774 336
	oct	4 387 263	122 161 599

CALCULATE

Cumulated total (in Time Series)

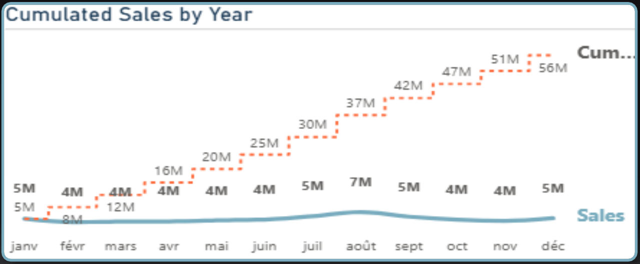
Cumulated Sales by Year

A variation of the previous model, but we have clarified the scope of the calculation (year).

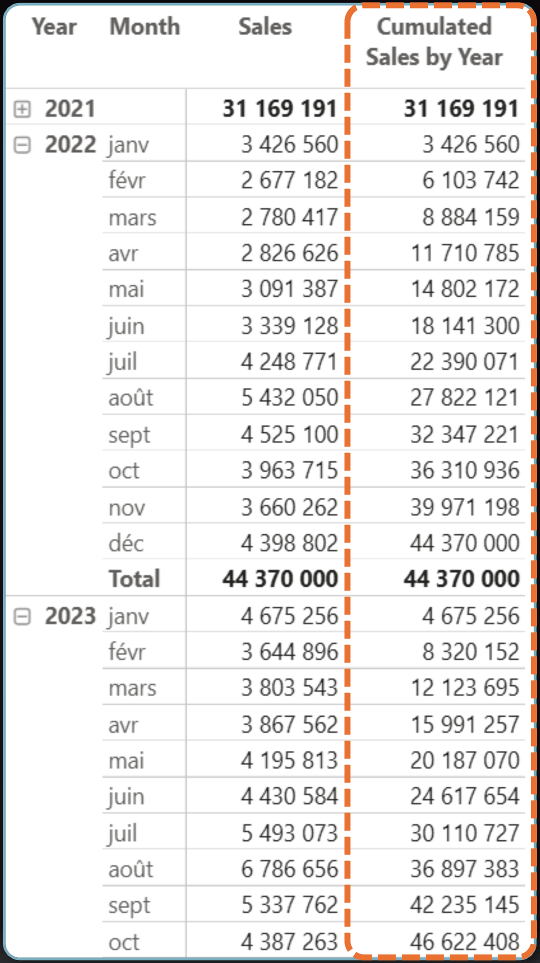


```
Cumulated Sales by Year =
VAR SelectedYear = SELECTEDVALUE('Dim Date'[Year])

RETURN
CALCULATE(
    [Sales],
        'Dim Date'[Date]<=MAX('Dim Date'[Date]),
        'Dim Date'[Year]= SelectedYear
)
```

Each year the cumulative sales start again.



Cumulated Sales by Year

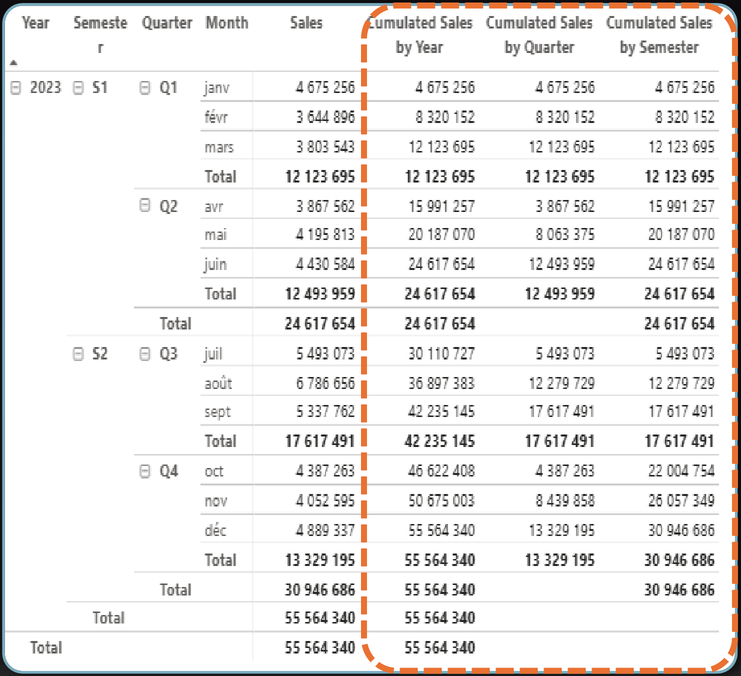


Year	Month	Sales	Cumulated Sales by Year
⊞ **2021**		**31 169 191**	**31 169 191**
⊟ **2022**	janv	3 426 560	3 426 560
	févr	2 677 182	6 103 742
	mars	2 780 417	8 884 159
	avr	2 826 626	11 710 785
	mai	3 091 387	14 802 172
	juin	3 339 128	18 141 300
	juil	4 248 771	22 390 071
	août	5 432 050	27 822 121
	sept	4 525 100	32 347 221
	oct	3 963 715	36 310 936
	nov	3 660 262	39 971 198
	déc	4 398 802	44 370 000
	Total	**44 370 000**	**44 370 000**
⊟ **2023**	janv	4 675 256	4 675 256
	févr	3 644 896	8 320 152
	mars	3 803 543	12 123 695
	avr	3 867 562	15 991 257
	mai	4 195 813	20 187 070
	juin	4 430 584	24 617 654
	juil	5 493 073	30 110 727
	août	6 786 656	36 897 383
	sept	5 337 762	42 235 145
	oct	4 387 263	46 622 408

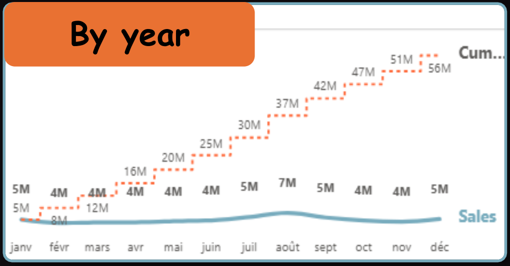
CALCULATE
Cumulated total (in Time Series)

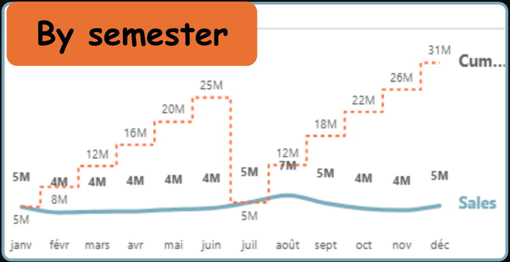
Cumulated Sales variations

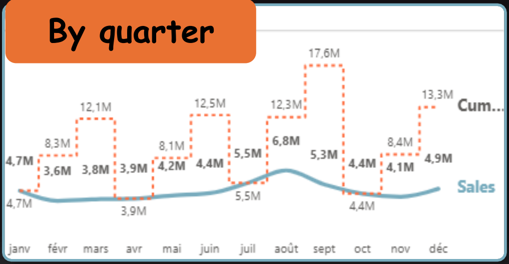
→ Many variations can be made, by quarter, by semester...



Year	Semester	Quarter	Month	Sales	Cumulated Sales by Year	Cumulated Sales by Quarter	Cumulated Sales by Semester
⊟ 2023	⊟ S1	⊟ Q1	janv	4 675 256	4 675 256	4 675 256	4 675 256
			févr	3 644 896	8 320 152	8 320 152	8 320 152
			mars	3 803 543	12 123 695	12 123 695	12 123 695
			Total	**12 123 695**	**12 123 695**	**12 123 695**	**12 123 695**
		⊟ Q2	avr	3 867 562	15 991 257	3 867 562	15 991 257
			mai	4 195 813	20 187 070	8 063 375	20 187 070
			juin	4 430 584	24 617 654	12 493 959	24 617 654
			Total	**12 493 959**	**24 617 654**	**12 493 959**	**24 617 654**
		Total		**24 617 654**	**24 617 654**		**24 617 654**
	⊟ S2	⊟ Q3	juil	5 493 073	30 110 727	5 493 073	5 493 073
			août	6 786 656	36 897 383	12 279 729	12 279 729
			sept	5 337 762	42 235 145	17 617 491	17 617 491
			Total	**17 617 491**	**42 235 145**	**17 617 491**	**17 617 491**
		⊟ Q4	oct	4 387 263	46 622 408	4 387 263	22 004 754
			nov	4 052 595	50 675 003	8 439 858	26 057 349
			déc	4 889 337	55 564 340	13 329 195	30 946 686
			Total	**13 329 195**	**55 564 340**	**13 329 195**	**30 946 686**
		Total		**30 946 686**	**55 564 340**		**30 946 686**
	Total			**55 564 340**	**55 564 340**		
Total				**55 564 340**	**55 564 340**		





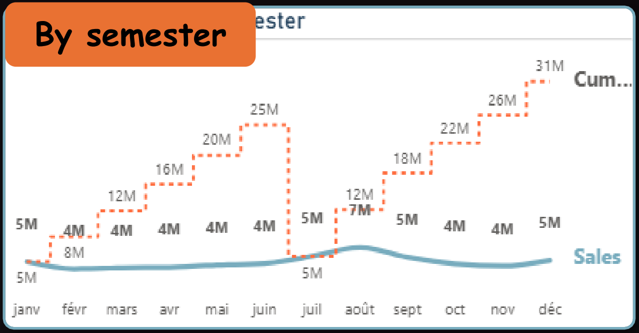


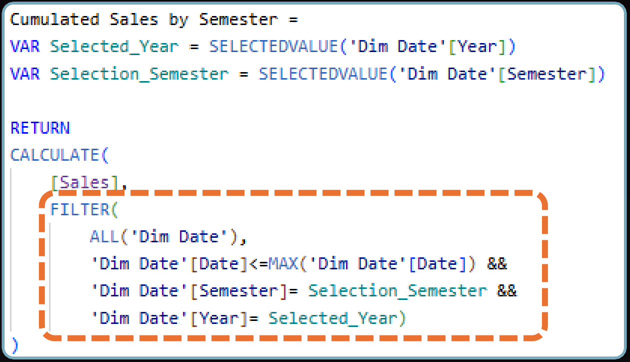
CALCULATE

Cumulated total (in Time Series)

Cumulated Sales variations

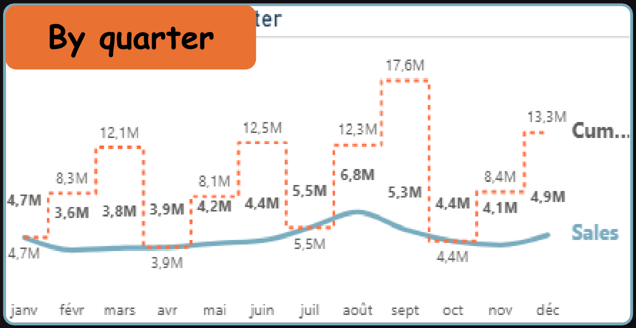
By semester

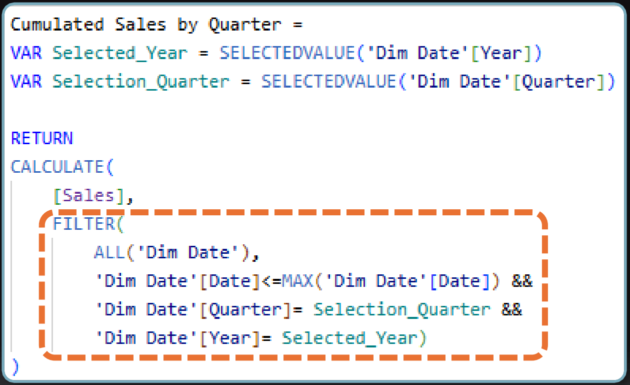




```
Cumulated Sales by Semester =
VAR Selected_Year = SELECTEDVALUE('Dim Date'[Year])
VAR Selection_Semester = SELECTEDVALUE('Dim Date'[Semester])

RETURN
CALCULATE(
    [Sales],
    FILTER(
        ALL('Dim Date'),
        'Dim Date'[Date]<=MAX('Dim Date'[Date]) &&
        'Dim Date'[Semester]= Selection_Semester &&
        'Dim Date'[Year]= Selected_Year)
)
```

By quarter



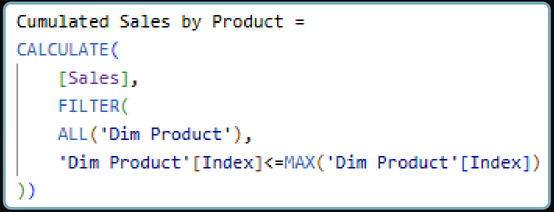


```
Cumulated Sales by Quarter =
VAR Selected_Year = SELECTEDVALUE('Dim Date'[Year])
VAR Selection_Quarter = SELECTEDVALUE('Dim Date'[Quarter])

RETURN
CALCULATE(
    [Sales],
    FILTER(
        ALL('Dim Date'),
        'Dim Date'[Date]<=MAX('Dim Date'[Date]) &&
        'Dim Date'[Quarter]= Selection_Quarter &&
        'Dim Date'[Year]= Selected_Year)
)
```

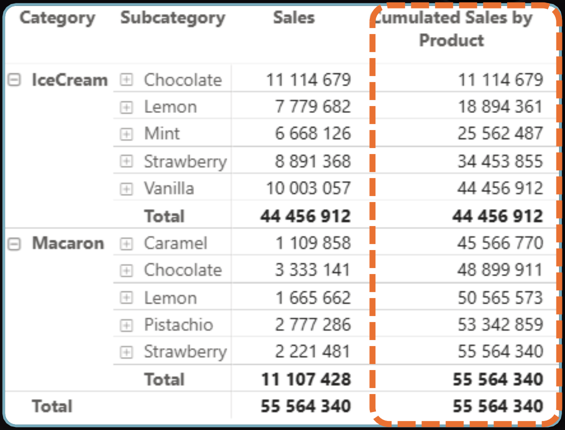
CALCULATE
Cumulated total (with Product dimension)

Cumulated Sales by Product

Here dimension is not a period but p product. It could be made with the dimensions Customer, Supplier, Income Statement, etc.

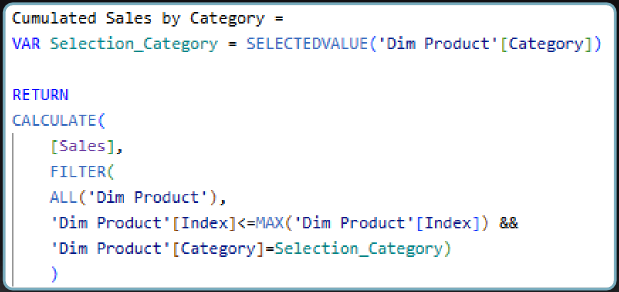


```
Cumulated Sales by Product =
CALCULATE(
    [Sales],
    FILTER(
    ALL('Dim Product'),
    'Dim Product'[Index]<=MAX('Dim Product'[Index])
))
```

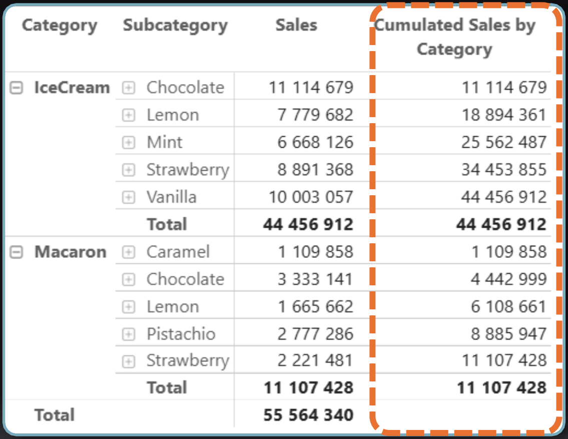


Category	Subcategory	Sales	Cumulated Sales by Product
⊟ IceCream	⊞ Chocolate	11 114 679	11 114 679
	⊞ Lemon	7 779 682	18 894 361
	⊞ Mint	6 668 126	25 562 487
	⊞ Strawberry	8 891 368	34 453 855
	⊞ Vanilla	10 003 057	44 456 912
	Total	**44 456 912**	**44 456 912**
⊟ Macaron	⊞ Caramel	1 109 858	45 566 770
	⊞ Chocolate	3 333 141	48 899 911
	⊞ Lemon	1 665 662	50 565 573
	⊞ Pistachio	2 777 286	53 342 859
	⊞ Strawberry	2 221 481	55 564 340
	Total	**11 107 428**	**55 564 340**
Total		**55 564 340**	**55 564 340**

Cumulated Sales by Category



```
Cumulated Sales by Category =
VAR Selection_Category = SELECTEDVALUE('Dim Product'[Category])

RETURN
CALCULATE(
    [Sales],
    FILTER(
    ALL('Dim Product'),
    'Dim Product'[Index]<=MAX('Dim Product'[Index]) &&
    'Dim Product'[Category]=Selection_Category)
    )
```

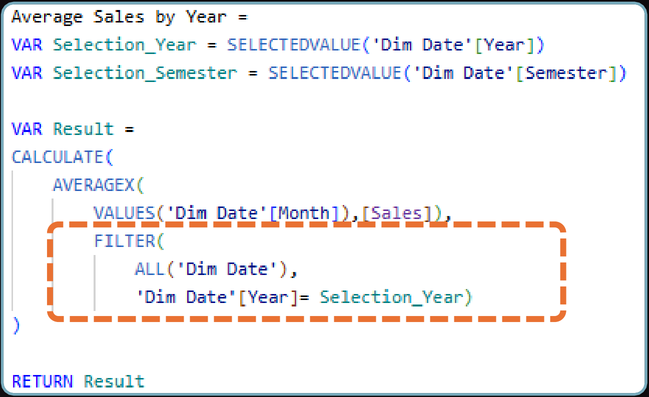


Category	Subcategory	Sales	Cumulated Sales by Category
⊟ IceCream	⊞ Chocolate	11 114 679	11 114 679
	⊞ Lemon	7 779 682	18 894 361
	⊞ Mint	6 668 126	25 562 487
	⊞ Strawberry	8 891 368	34 453 855
	⊞ Vanilla	10 003 057	44 456 912
	Total	**44 456 912**	**44 456 912**
⊟ Macaron	⊞ Caramel	1 109 858	1 109 858
	⊞ Chocolate	3 333 141	4 442 999
	⊞ Lemon	1 665 662	6 108 661
	⊞ Pistachio	2 777 286	8 885 947
	⊞ Strawberry	2 221 481	11 107 428
	Total	**11 107 428**	**11 107 428**
Total		**55 564 340**	

CALCULATE
Average (in Time Series)

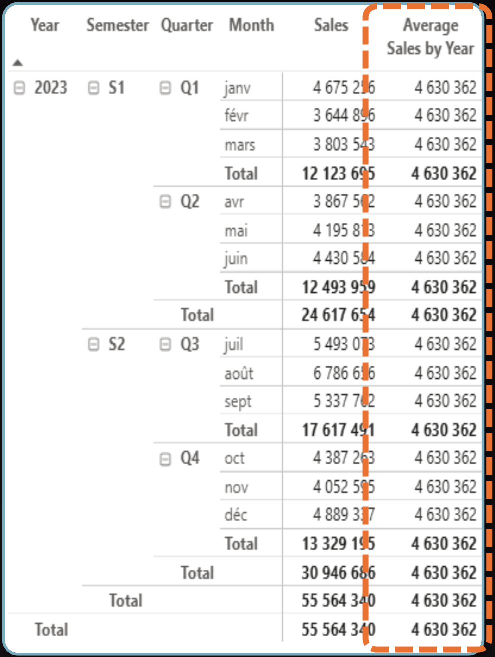
Average Sales by Year

A variation of the previous model, but we have clarified the scope of the calculation (year).

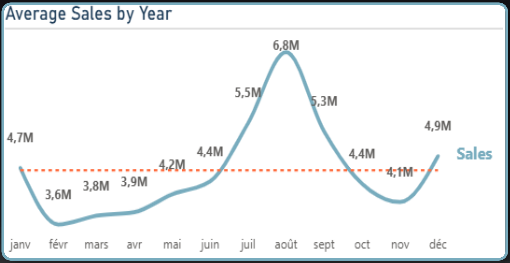


```
Average Sales by Year =
VAR Selection_Year = SELECTEDVALUE('Dim Date'[Year])
VAR Selection_Semester = SELECTEDVALUE('Dim Date'[Semester])

VAR Result =
CALCULATE(
    AVERAGEX(
        VALUES('Dim Date'[Month]),[Sales]),
        FILTER(
            ALL('Dim Date'),
            'Dim Date'[Year]= Selection_Year)
)

RETURN Result
```

AVERAGEX creates favorable conditions for iteration in association with VALUES. Each year, the average sales are recalculated using the FILTER pattern.



Year	Semester	Quarter	Month	Sales	Average Sales by Year
⊟ 2023	⊟ S1	⊟ Q1	janv	4 675 256	4 630 362
			févr	3 644 896	4 630 362
			mars	3 803 543	4 630 362
			Total	**12 123 695**	**4 630 362**
		⊟ Q2	avr	3 867 562	4 630 362
			mai	4 195 813	4 630 362
			juin	4 430 584	4 630 362
			Total	**12 493 959**	**4 630 362**
		Total		**24 617 654**	**4 630 362**
	⊟ S2	⊟ Q3	juil	5 493 073	4 630 362
			août	6 786 656	4 630 362
			sept	5 337 762	4 630 362
			Total	**17 617 491**	**4 630 362**
		⊟ Q4	oct	4 387 263	4 630 362
			nov	4 052 595	4 630 362
			déc	4 889 337	4 630 362
			Total	**13 329 195**	**4 630 362**
		Total		**30 946 686**	**4 630 362**
	Total			**55 564 340**	**4 630 362**
Total				**55 564 340**	**4 630 362**



Start with PowerBI

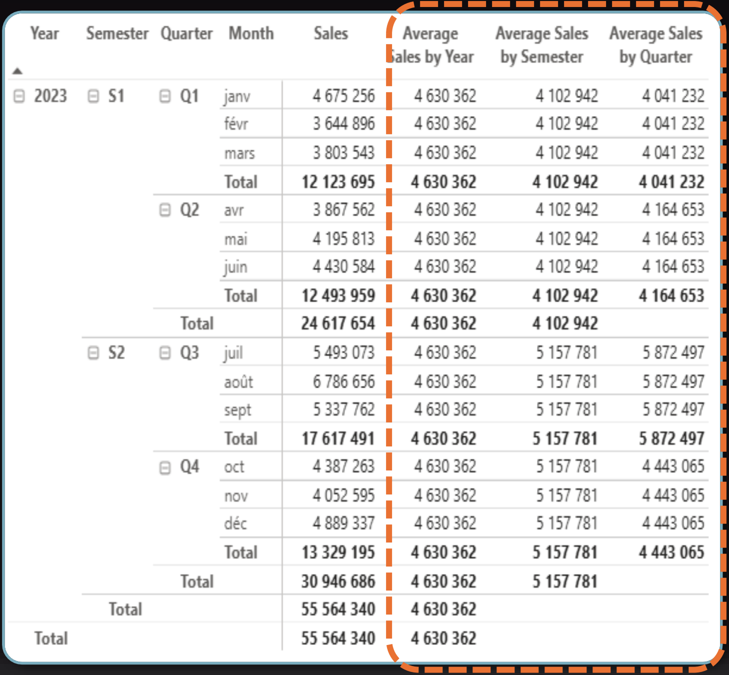
"From Rookie to Rock"

Patou Tips #48 – Quick Guide of Calculate

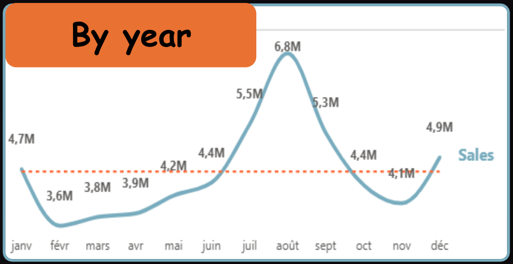
CALCULATE
Average (in Time Series)

Average Sales variations

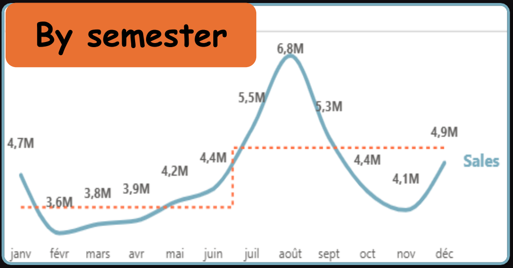
→ This model is really interesting because many variations can be made, by quarter, by semester...



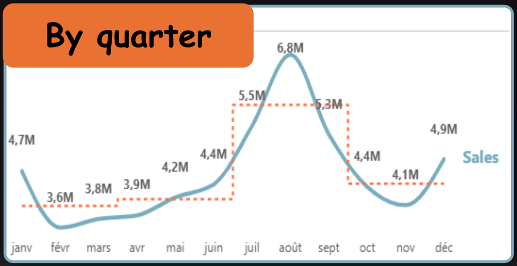
Year	Semester	Quarter	Month	Sales	Average Sales by Year	Average Sales by Semester	Average Sales by Quarter
⊟ 2023	⊟ S1	⊟ Q1	janv	4 675 256	4 630 362	4 102 942	4 041 232
			févr	3 644 896	4 630 362	4 102 942	4 041 232
			mars	3 803 543	4 630 362	4 102 942	4 041 232
			Total	**12 123 695**	**4 630 362**	**4 102 942**	**4 041 232**
		⊟ Q2	avr	3 867 562	4 630 362	4 102 942	4 164 653
			mai	4 195 813	4 630 362	4 102 942	4 164 653
			juin	4 430 584	4 630 362	4 102 942	4 164 653
			Total	**12 493 959**	**4 630 362**	**4 102 942**	**4 164 653**
		Total		**24 617 654**	**4 630 362**	**4 102 942**	
	⊟ S2	⊟ Q3	juil	5 493 073	4 630 362	5 157 781	5 872 497
			août	6 786 656	4 630 362	5 157 781	5 872 497
			sept	5 337 762	4 630 362	5 157 781	5 872 497
			Total	**17 617 491**	**4 630 362**	**5 157 781**	**5 872 497**
		⊟ Q4	oct	4 387 263	4 630 362	5 157 781	4 443 065
			nov	4 052 595	4 630 362	5 157 781	4 443 065
			déc	4 889 337	4 630 362	5 157 781	4 443 065
			Total	**13 329 195**	**4 630 362**	**5 157 781**	**4 443 065**
		Total		**30 946 686**	**4 630 362**	**5 157 781**	
	Total			**55 564 340**	**4 630 362**		
Total				**55 564 340**	**4 630 362**		



By year



By semester

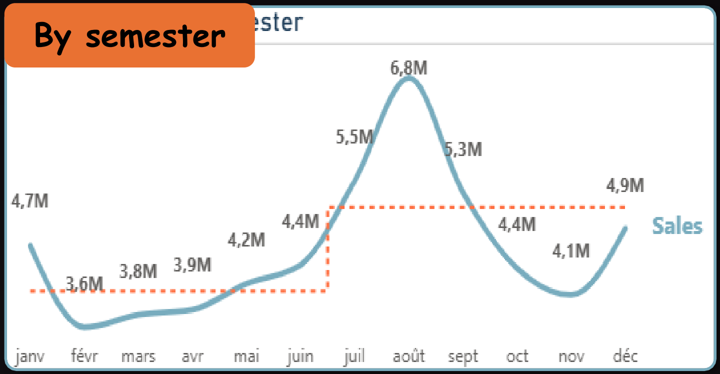


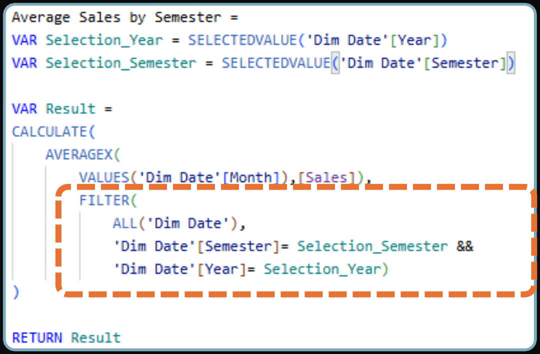
By quarter

Start with PowerBI
"From Rookie to Rock"

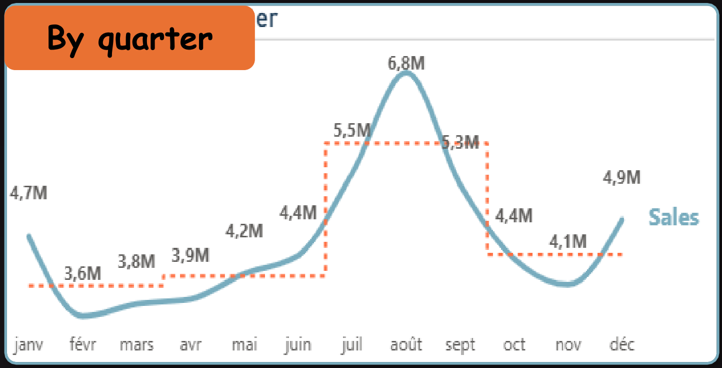
CALCULATE
Average (in Time Series)

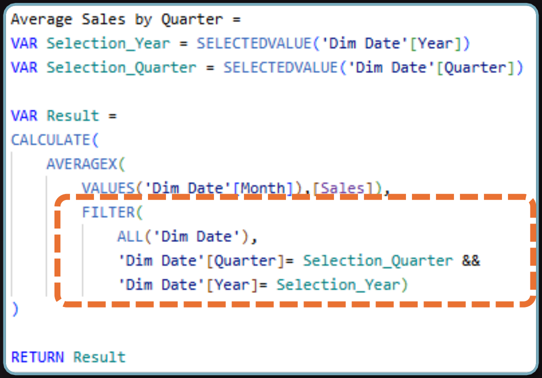
Average Sales variations


By semester



```
Average Sales by Semester =
VAR Selection_Year = SELECTEDVALUE('Dim Date'[Year])
VAR Selection_Semester = SELECTEDVALUE('Dim Date'[Semester])

VAR Result =
CALCULATE(
    AVERAGEX(
        VALUES('Dim Date'[Month]),[Sales]),
        FILTER(
            ALL('Dim Date'),
            'Dim Date'[Semester]= Selection_Semester &&
            'Dim Date'[Year]= Selection_Year)
)

RETURN Result
```


By quarter



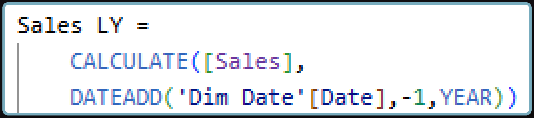
```
Average Sales by Quarter =
VAR Selection_Year = SELECTEDVALUE('Dim Date'[Year])
VAR Selection_Quarter = SELECTEDVALUE('Dim Date'[Quarter])

VAR Result =
CALCULATE(
    AVERAGEX(
        VALUES('Dim Date'[Month]),[Sales]),
        FILTER(
            ALL('Dim Date'),
            'Dim Date'[Quarter]= Selection_Quarter &&
            'Dim Date'[Year]= Selection_Year)
)

RETURN Result
```

CALCULATE
Time Intelligence

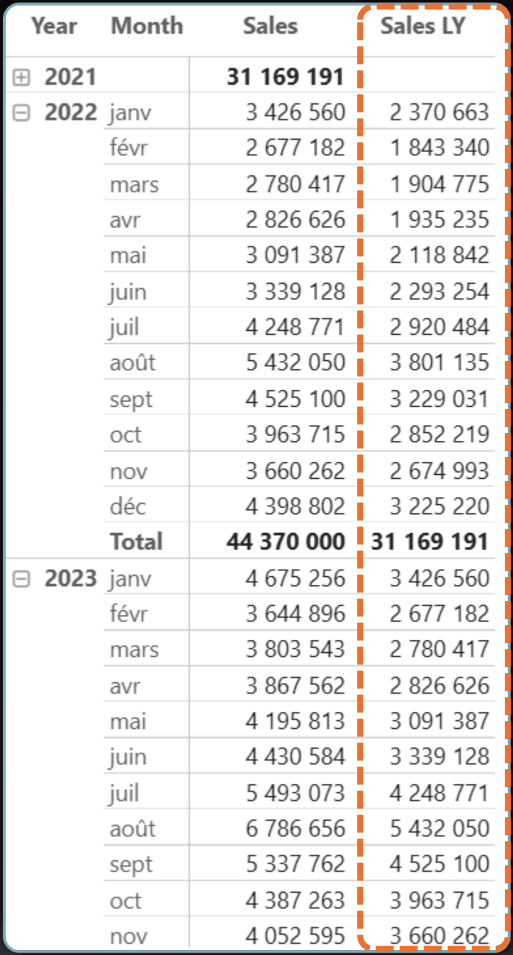
Sales Last Year (LY)

Thanks to the time calculation function, it's the perfect winning combination.

A function such "DATEADD" allow you a lot patterns



```
Sales LY =
    CALCULATE([Sales],
    DATEADD('Dim Date'[Date],-1,YEAR))
```



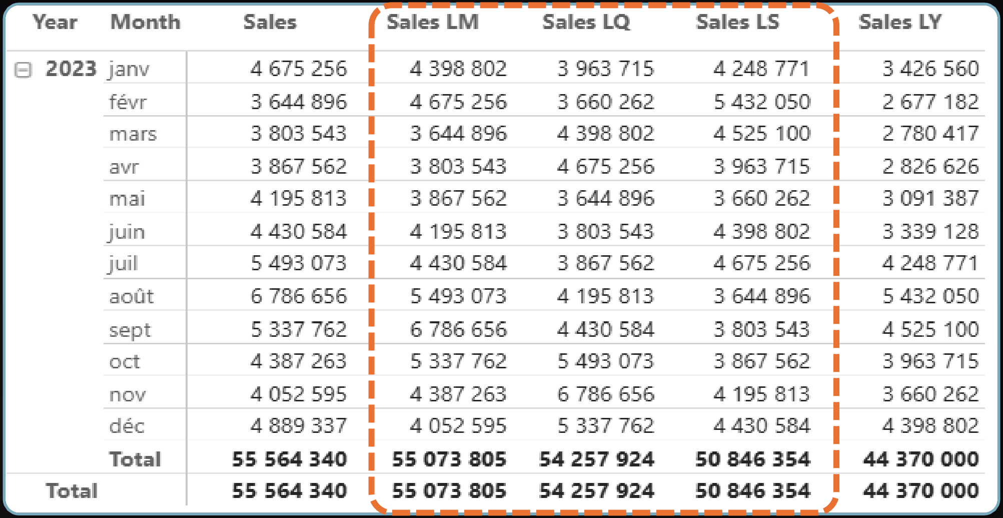
Year	Month	Sales	Sales LY
⊞ 2021		31 169 191	
⊟ 2022	janv	3 426 560	2 370 663
	févr	2 677 182	1 843 340
	mars	2 780 417	1 904 775
	avr	2 826 626	1 935 235
	mai	3 091 387	2 118 842
	juin	3 339 128	2 293 254
	juil	4 248 771	2 920 484
	août	5 432 050	3 801 135
	sept	4 525 100	3 229 031
	oct	3 963 715	2 852 219
	nov	3 660 262	2 674 993
	déc	4 398 802	3 225 220
	Total	**44 370 000**	**31 169 191**
⊟ 2023	janv	4 675 256	3 426 560
	févr	3 644 896	2 677 182
	mars	3 803 543	2 780 417
	avr	3 867 562	2 826 626
	mai	4 195 813	3 091 387
	juin	4 430 584	3 339 128
	juil	5 493 073	4 248 771
	août	6 786 656	5 432 050
	sept	5 337 762	4 525 100
	oct	4 387 263	3 963 715
	nov	4 052 595	3 660 262

Start with PowerBI

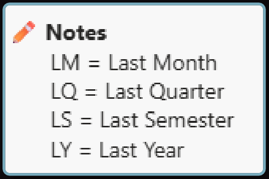
"From Rookie to Rock"

CALCULATE
Time Intelligence

Sales Last Period



Year	Month	Sales	Sales LM	Sales LQ	Sales LS	Sales LY
⊟ **2023**	janv	4 675 256	4 398 802	3 963 715	4 248 771	3 426 560
	févr	3 644 896	4 675 256	3 660 262	5 432 050	2 677 182
	mars	3 803 543	3 644 896	4 398 802	4 525 100	2 780 417
	avr	3 867 562	3 803 543	4 675 256	3 963 715	2 826 626
	mai	4 195 813	3 867 562	3 644 896	3 660 262	3 091 387
	juin	4 430 584	4 195 813	3 803 543	4 398 802	3 339 128
	juil	5 493 073	4 430 584	3 867 562	4 675 256	4 248 771
	août	6 786 656	5 493 073	4 195 813	3 644 896	5 432 050
	sept	5 337 762	6 786 656	4 430 584	3 803 543	4 525 100
	oct	4 387 263	5 337 762	5 493 073	3 867 562	3 963 715
	nov	4 052 595	4 387 263	6 786 656	4 195 813	3 660 262
	déc	4 889 337	4 052 595	5 337 762	4 430 584	4 398 802
	Total	**55 564 340**	**55 073 805**	**54 257 924**	**50 846 354**	**44 370 000**
Total		**55 564 340**	**55 073 805**	**54 257 924**	**50 846 354**	**44 370 000**

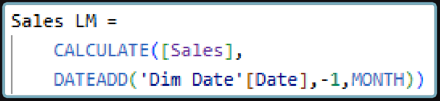


Notes

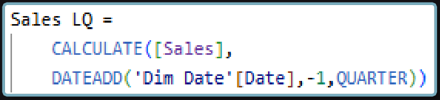
LM = Last Month
LQ = Last Quarter
LS = Last Semester
LY = Last Year

Many variations can be made, by week, by month, by quarter, by semester...

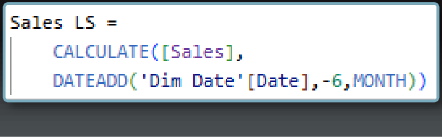
For each selected period, the previous value for that period is displayed.



```
Sales LM =
    CALCULATE([Sales],
    DATEADD('Dim Date'[Date],-1,MONTH))
```



```
Sales LQ =
    CALCULATE([Sales],
    DATEADD('Dim Date'[Date],-1,QUARTER))
```



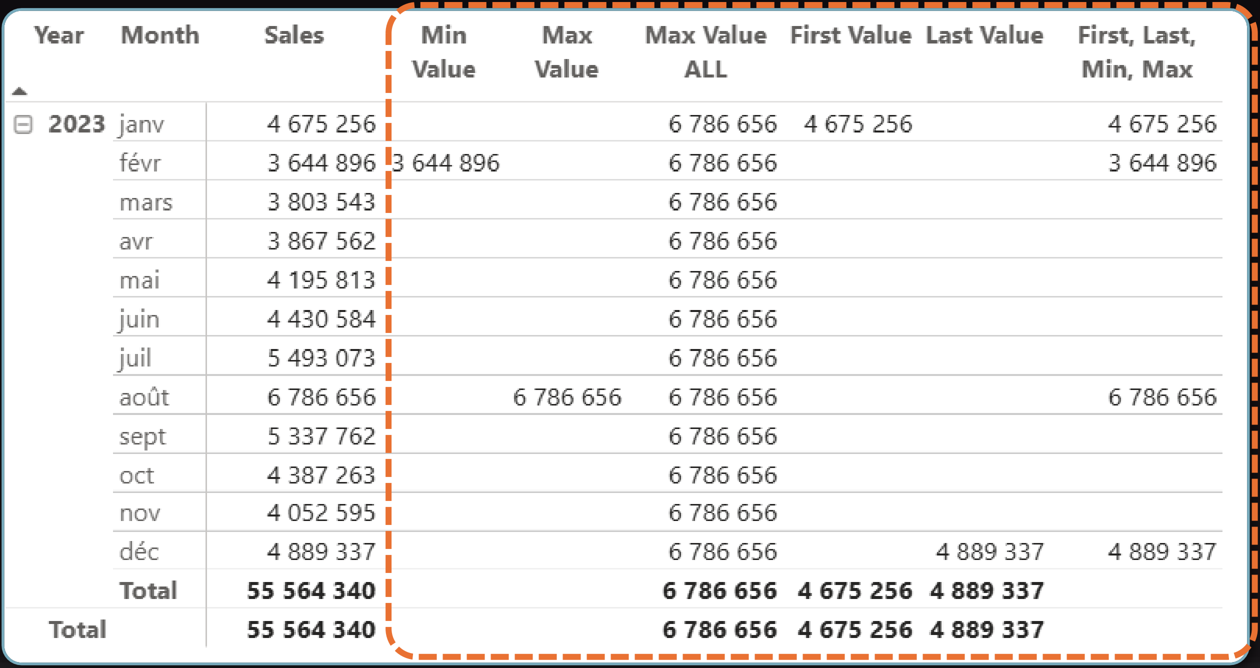
```
Sales LS =
    CALCULATE([Sales],
    DATEADD('Dim Date'[Date],-6,MONTH))
```

CALCULATE

Min, Max, First, Last values

Min, Max, First, Last Sales

Often, certain specific values are needed for later uses, for formulas, but also, as in the examples to come, to improve a visualization.



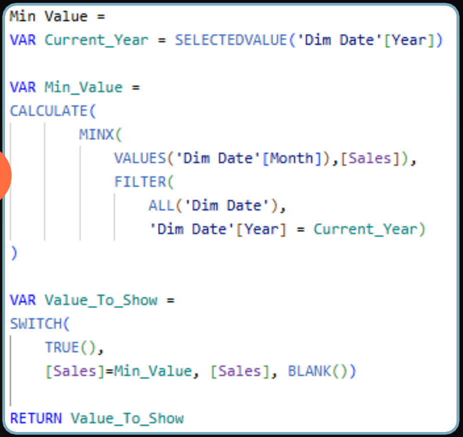
Year	Month	Sales	Min Value	Max Value	Max Value ALL	First Value	Last Value	First, Last, Min, Max
2023	janv	4 675 256			6 786 656	4 675 256		4 675 256
	févr	3 644 896	3 644 896		6 786 656			3 644 896
	mars	3 803 543			6 786 656			
	avr	3 867 562			6 786 656			
	mai	4 195 813			6 786 656			
	juin	4 430 584			6 786 656			
	juil	5 493 073			6 786 656			
	août	6 786 656		6 786 656	6 786 656			6 786 656
	sept	5 337 762			6 786 656			
	oct	4 387 263			6 786 656			
	nov	4 052 595			6 786 656			
	déc	4 889 337			6 786 656		4 889 337	4 889 337
	Total	**55 564 340**			**6 786 656**	**4 675 256**	**4 889 337**	
Total		**55 564 340**			**6 786 656**	**4 675 256**	**4 889 337**	

CALCULATE

Min, Max, First, Last values

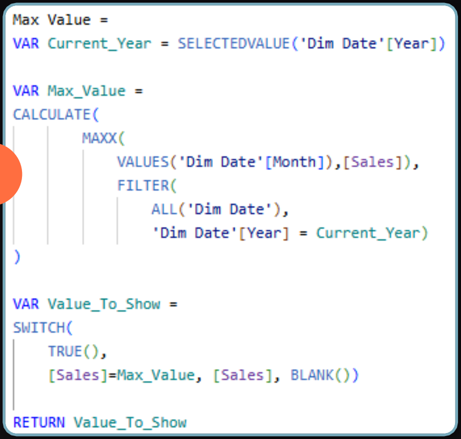
Min, Max

1



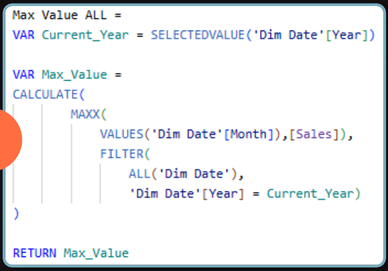
```
Min Value =
VAR Current_Year = SELECTEDVALUE('Dim Date'[Year])

VAR Min_Value =
CALCULATE(
        MINX(
            VALUES('Dim Date'[Month]),[Sales]),
            FILTER(
                ALL('Dim Date'),
                'Dim Date'[Year] = Current_Year)
)

VAR Value_To_Show =
SWITCH(
    TRUE(),
    [Sales]=Min_Value, [Sales], BLANK())

RETURN Value_To_Show
```

2



```
Max Value =
VAR Current_Year = SELECTEDVALUE('Dim Date'[Year])

VAR Max_Value =
CALCULATE(
        MAXX(
            VALUES('Dim Date'[Month]),[Sales]),
            FILTER(
                ALL('Dim Date'),
                'Dim Date'[Year] = Current_Year)
)

VAR Value_To_Show =
SWITCH(
    TRUE(),
    [Sales]=Max_Value, [Sales], BLANK())

RETURN Value_To_Show
```

3



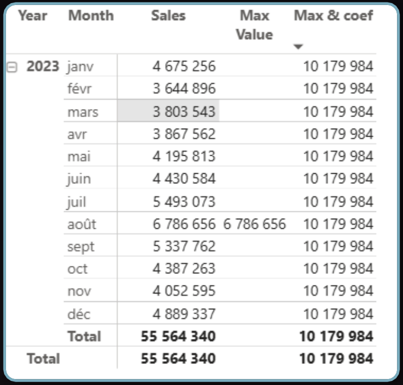
```
Max Value ALL =
VAR Current_Year = SELECTEDVALUE('Dim Date'[Year])

VAR Max_Value =
CALCULATE(
        MAXX(
            VALUES('Dim Date'[Month]),[Sales]),
            FILTER(
                ALL('Dim Date'),
                'Dim Date'[Year] = Current_Year)
)

RETURN Max_Value
```

1 2 The MINX and MAXX functions create favorable conditions for iteration in with VALUES. Then, only the desired value (Min or Max) is displayed.

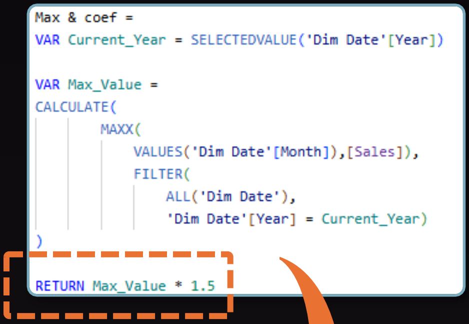
3 With ALL, it might be useful to populate the Min or Max value across the entire time series (see next page).

CALCULATE
Min, Max, First, Last values

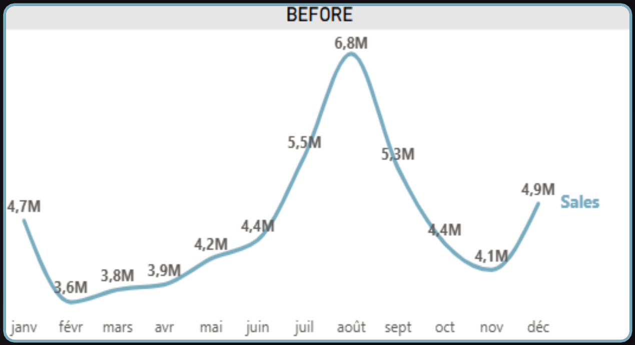
Min, Max

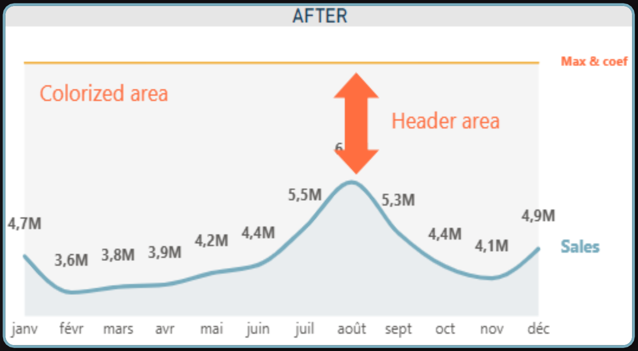


Year	Month	Sales	Max Value	Max & coef
⊟ 2023	janv	4 675 256		10 179 984
	févr	3 644 896		10 179 984
	mars	3 803 543		10 179 984
	avr	3 867 562		10 179 984
	mai	4 195 813		10 179 984
	juin	4 430 584		10 179 984
	juil	5 493 073		10 179 984
	août	6 786 656	6 786 656	10 179 984
	sept	5 337 762		10 179 984
	oct	4 387 263		10 179 984
	nov	4 052 595		10 179 984
	déc	4 889 337		10 179 984
	Total	**55 564 340**		**10 179 984**
Total		**55 564 340**		**10 179 984**



```
Max & coef =
VAR Current_Year = SELECTEDVALUE('Dim Date'[Year])

VAR Max_Value =
CALCULATE(
        MAXX(
            VALUES('Dim Date'[Month]),[Sales]),
            FILTER(
                ALL('Dim Date'),
                'Dim Date'[Year] = Current_Year)
)

RETURN Max_Value * 1.5
```

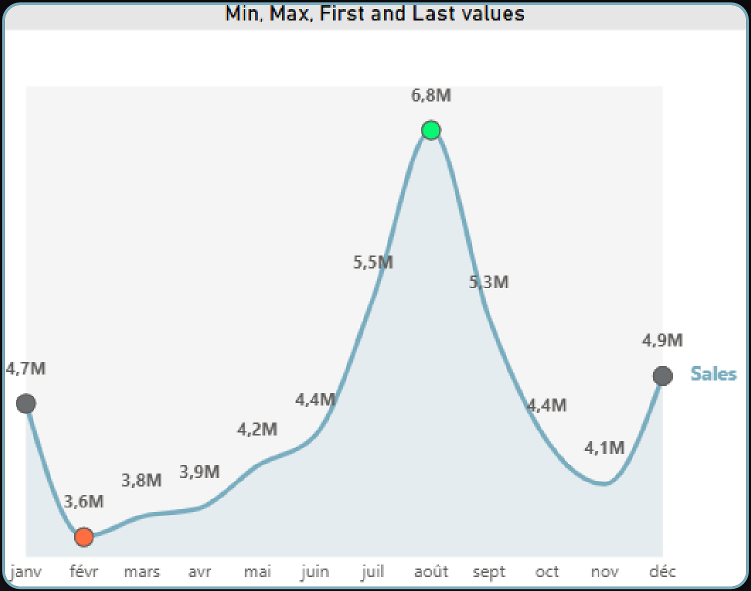


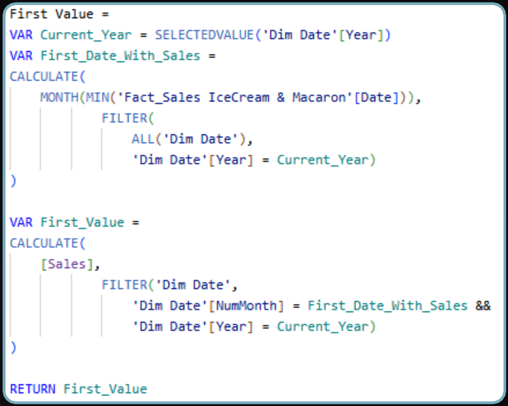


Using a coefficient, the ALL calculation allows you to place a line (or an area) above your graph with the Max value or below it with the Min value. This is a useful trick to improve your visualization.

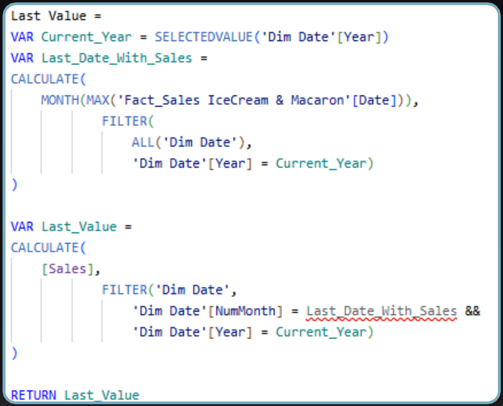
CALCULATE
Min, Max, First, Last values

Min, Max, First, last





```
First Value =
VAR Current_Year = SELECTEDVALUE('Dim Date'[Year])
VAR First_Date_With_Sales =
CALCULATE(
    MONTH(MIN('Fact_Sales IceCream & Macaron'[Date])),
            FILTER(
                ALL('Dim Date'),
                'Dim Date'[Year] = Current_Year)
)

VAR First_Value =
CALCULATE(
    [Sales],
            FILTER('Dim Date',
                'Dim Date'[NumMonth] = First_Date_With_Sales &&
                'Dim Date'[Year] = Current_Year)
)

RETURN First_Value
```



```
Last Value =
VAR Current_Year = SELECTEDVALUE('Dim Date'[Year])
VAR Last_Date_With_Sales =
CALCULATE(
    MONTH(MAX('Fact_Sales IceCream & Macaron'[Date])),
            FILTER(
                ALL('Dim Date'),
                'Dim Date'[Year] = Current_Year)
)

VAR Last_Value =
CALCULATE(
    [Sales],
            FILTER('Dim Date',
                'Dim Date'[NumMonth] = Last_Date_With_Sales &&
                'Dim Date'[Year] = Current_Year)
)

RETURN Last_Value
```

It might be interesting to also add the first or last value of a time series.

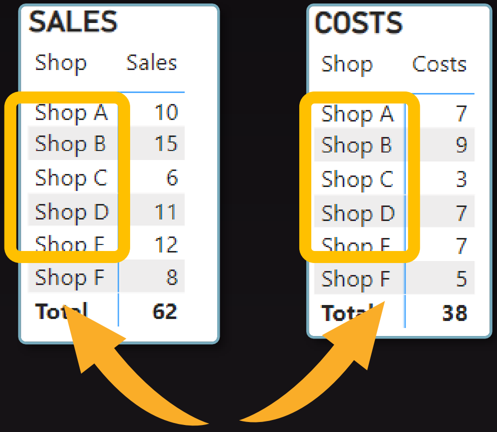
CALCULATE

Use Treatas to connect 2 different data sources

When to use it? To perform a calculation with a variable from another table. Thanks to the Treatas formula it can be more interesting to create a virtual link if one or more references are common.

Example: we have two different data sources (Sales and Costs) and we want to calculate the margin.

2 tables and 2 data sources



SALES

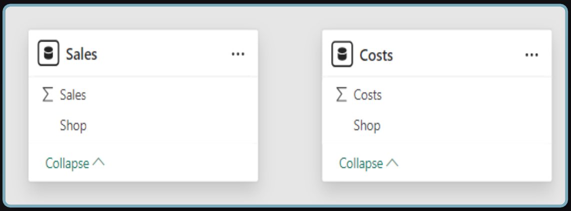
Shop	Sales
Shop A	10
Shop B	15
Shop C	6
Shop D	11
Shop E	12
Shop F	8
Total	62

COSTS

Shop	Costs
Shop A	7
Shop B	9
Shop C	3
Shop D	7
Shop E	7
Shop F	5
Total	38

Common references

No connection between the two data sources



CALCULATE

Use Treatas to connect 2 different data sources

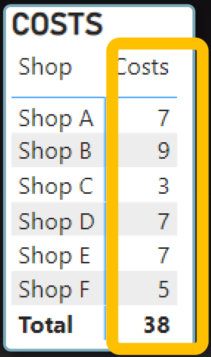
Treat the columns of the input table (SALES) as column from other table (COSTS).

```
Costs with Treatas =
CALCULATE(SUM(Costs[Costs]),
    TREATAS(VALUES(Sales[Shop]),Costs[Shop])
)
```

Input table

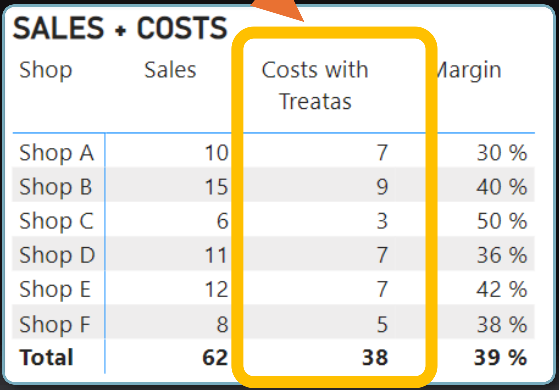
Other table

Note, VALUES search all the unique values from the input table. This pattern is really usefull.



COSTS

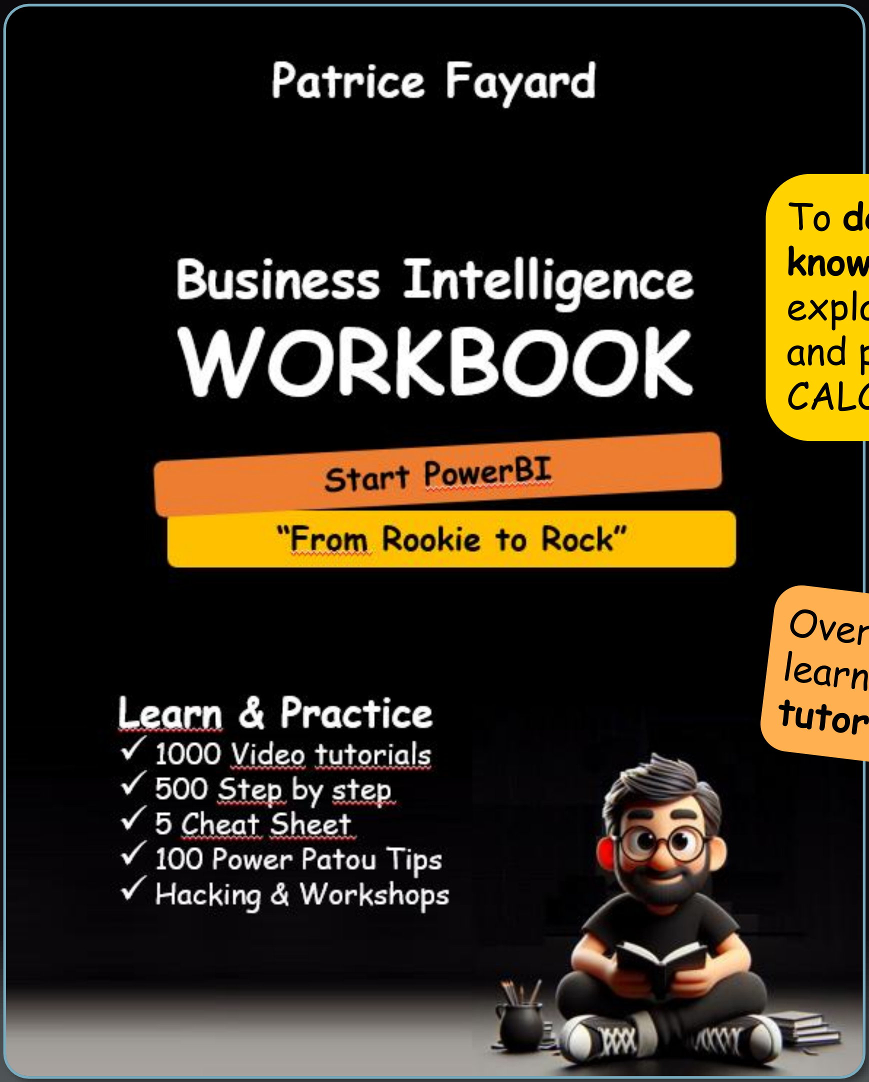
Shop	Costs
Shop A	7
Shop B	9
Shop C	3
Shop D	7
Shop E	7
Shop F	5
Total	**38**



SALES + COSTS

Shop	Sales	Costs with Treatas	Margin
Shop A	10	7	30 %
Shop B	15	9	40 %
Shop C	6	3	50 %
Shop D	11	7	36 %
Shop E	12	7	42 %
Shop F	8	5	38 %
Total	**62**	**38**	**39 %**

Coming soon, in 2026!



Patrice Fayard

Business Intelligence

WORKBOOK

Start PowerBI

"From Rookie to Rock"

Learn & Practice

✓ 1000 Video tutorials

✓ 500 Step by step

✓ 5 Cheat Sheet

✓ 100 Power Patou Tips

✓ Hacking & Workshops

To **develop your knowledge**, find more explanations, exercises and patterns on CALCULATE.

Over **500 pages** to learn and practice: **video tutorials and resources**

Learn and practice

Find past issues of "Patou Tips" and download resources to practice on GitHub

Easy to do it...

Patou Tips #5

Create a Customized Chart
(for income statement)

To practice
downloadable free
resources in GitHub

Patou Tips #5
Create a Customized Chart
(for income statement)

Easy to do it...

Patou Tips #6

Create
Customized Icon

(with PowerPoint
for PowerBI)

Patou Tips #6
Create
Customized Icon

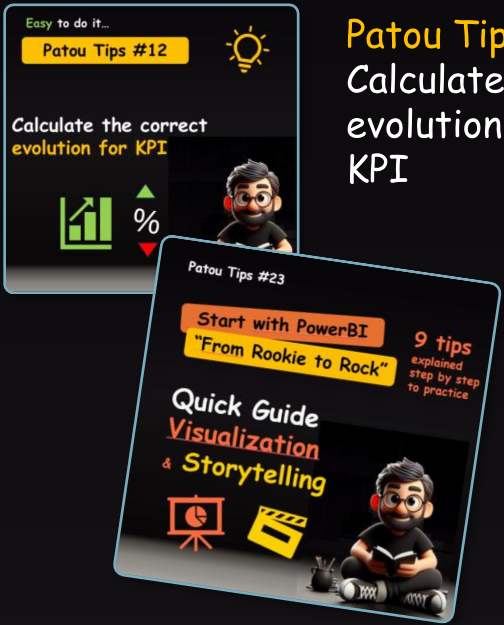
Easy to do it...

Patou Tips #7

Create an Age Pyramid Chart
(for Human Ressources)

To practice
downloadable free
resources in GitHub

Patou Tips #7
Create an Age
Pyramid Chart
(for Human
Ressources)



Easy to do it...

Patou Tips #12

Calculate the correct
evolution for KPI

Patou Tips #23

Start with PowerBI
"From Rookie to Rock"

9 tips
explained
step by step
to practice

Quick Guide
Visualization
& Storytelling

Patou Tips #12
Calculate right
evolution for
KPI

Patou Tips #23
Quick Guide Visualization &
Storytelling

Resources on GitHub
https://github.com/Patou
Tips/Patou-Tips

Don’t forget!
This isn't the truth, it's just my truth!

Patou Tips

Follow me
Like me
Share me