**Windownsum\_funcation**

windowSum function calculates the sum of the aggregated measure in a custom window that is partitioned and sorted by specified attributes. Usually, you use custom window functions on a time series, where your visual shows a metric and a date field.

**Syntax :**

windowSum

(

*measure*

, *[sort\_order\_field ASC/DESC, ...]*

, *start\_index*

, *end\_index*

,*[ partition\_field, ... ]*

)

**Example :**

example calculates the moving sum of sum(Revenue), sorted by SaleDate. The calculation includes two rows above and one row ahead of the current row.

windowSum

(

sum(Revenue),

[SaleDate ASC],

2,

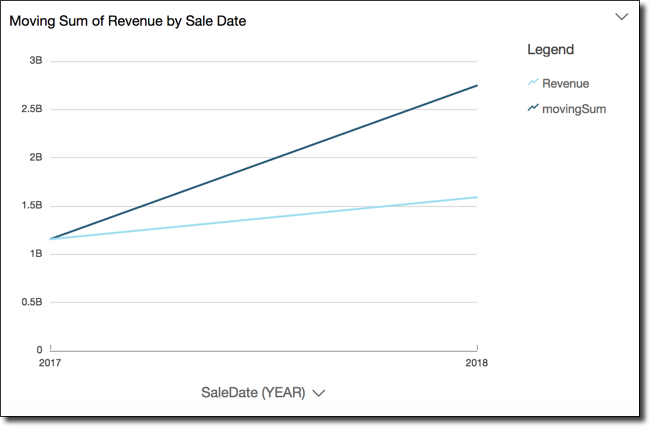
1

)

example show a trailing 12-month sum.

windowSum(sum(Revenue),[SaleDate ASC],12,0)

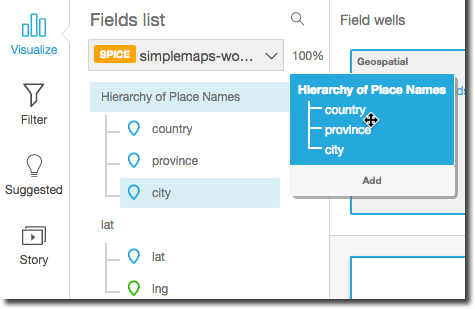
The sum(Revenue) field is added to the chart to show the difference between the revenue and the trailing 12-month sum of revenue.



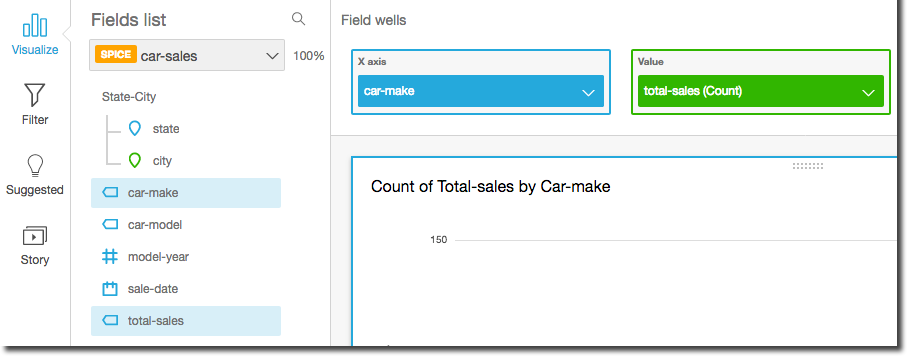
**drill\_downs\_funcationality :**

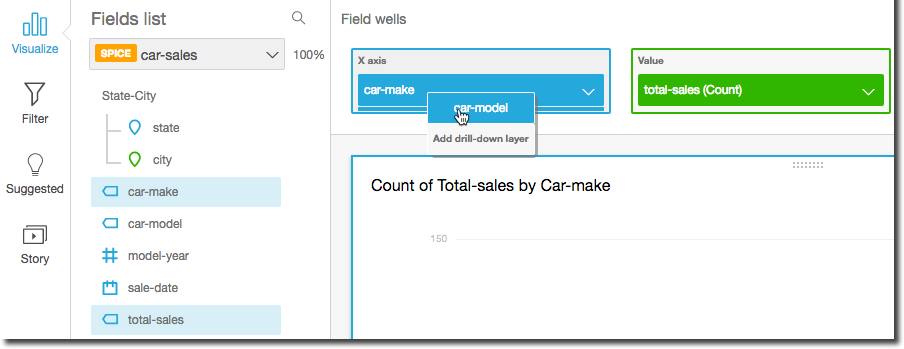
Drill-down functionality is added automatically for dates when you associate a date field with the drill-down field well of a visual. In this case, you can always drill up and down through the levels of date granularity. Drill-down functionality is also added automatically for geospatial groupings, after you define these in the dataset.

If your dataset has a defined hierarchy, you can drag the entire hierarchy into the field well as one. An example is geospatial or coordinate data.



the drill-down hierarchy to an appropriate field well, depending on the visual type. Make sure that the label for the dragged field says **Add drill-down layer**. Position the dragged field above or below the existing field based on where you want it to be in the hierarchy you're creating.





**Syntax :**

Week = if Date.IsInCurrentWeek([Date])

then "This Week"

else Text.From(Date.EndOfWeek([Date]))

forecast =

Var \_CYMonth = MONTH(TODAY())

Var \_RYMonth = if(\_CYMonth > 3 && \_CYMonth <= 12, \_CYMonth - 3,\_CYMonth + 9)

Var cum\_sum\_last\_month = CALCULATE([\*\*bleep\*\* NormValue],Dates[Date]<=EOMONTH(EOMONTH(TODAY(),0),0))

Var remain\_month\_value = CALCULATE(SUM('5y average'[Value]),FILTER(ALLSELECTED('5y average'),MAX('5y average'[Month Number])>=\_RYMonth&&AND('5y average'[Month Number]<=MAX('5y average'[Month Number]),'5y average'[Month Number]>=\_RYMonth)))

Return

IF(not(ISBLANK(MAX(Dates[RY Month Number]))) && MAX(Dates[RY Month Number])>=\_RYMonth,cum\_sum\_last\_month+remain\_month\_value,BLANK())