DAX → Calculate. ↳ Measure ↳ where (selection criterion in dataframe).

Calculate (Exp, Fiter 1, *Args)
- ↳ Measure → Previously created
- ↳ New measure exp. Sum

↳ Fiter one after another as args

Bulk order = Calculate ([Total_order], All_sales [OrderQuantity] > 1)

* Remember Split - Apply - Combine.

In groupby → On row_id fiter → Apply function → Concat / Combine.

** When we use calculate ÷ Split → Calculate fiter → Apply → Combine

df.groupby(['col1']) [ ].apply(lambda df: (df[col2] > 3).sum())

↳ df.groupby( ).filter( )

↳ How many bulk orders, Weekend orders.

ALL
↳ select all rows that satisfies the cond.

→ Diversion [select sum(col1) as Den, sum(col1) over(partition by col2) as N from df.]

↳ [df.groupby(col2)[col1].sum(), df[col1].sum()]
↳ concat
↳ All helps here

Revenue = Price * Quantity

All Orders = CALCULATE ([Total Orders], ALL (AW_Sales))

Exp = ([Total Orders] + 1 , Fiber * _ _)
↳ can be

↳ Let say we wish to evaluate percentage of sales per category
product
order quantity.
↳ groupby (['Product_cat']) ('OrderQuantity') . Sum() / df [OrderQuan] . Sum)

⇒ Measure (Sum (AW_Sales [Ord_Qu])) / Measure (Sum [AW_Sales [Order_Quan]]) , All [AW_Sales])

⇒ Ex-2 → We want to store overall average price
↳ calculate (Exp, with All
Calculate ([Avg (Prod_price)] , All (AW_Product))

Sum X → Apply axis 1, → On multiple rows.

sum (df [col1]) * sum (df [col2])

↳ sum (df [col1] * df [col2]) → Sum X
↳ Elementwise operation measures.

Revenue = Price * Quantity

↳ groupby Product find Revenue

↳ We need to create a measure for the same.

Mea = SUMX(AW_sales, Aw_sales[Price] * Aw_sale[Qty])

↳ Table ↳ Col1 * Col2

↳ SUMX(Aw_sales, Aw_sales[Quantity] * Related(Aw_Product[Price]))

Join on key and fetch column from some other table.

Datetime Manipulation.

Cumsum, lead_lag, rolling.

df.sort_values(['col1'])[col2].cumsum

↳ sum(Col2) over (order by col1)

df.sort_values(col2)

df.groupby(['col1']).[col3].transform(pd.series.cumsum)

lag(Col1, 1, 0) over (partion by Col2, order by col

df.sort_values(col3).groupby(col1).['col2'].shift()

↳ In built transform operation.

...to day Revenue.

YTD Revenue = CALCULATE([Total_Revenue], DATESYTD('AW_Calender - Lookup'[Date]))

↳ Powerful **

ERD

Join these 2 on the ~~ad~~ respective columns
↳ dates : Revenue. (for each order).

Group by extract(year from date), dt.month
dt.year
↳ sum(Revenue). ~~over(order by month, partition by year)~~

↳ cumsum(sa) ~~order by year~~.

Over(order by month partition by year).