

Business Intelligence with Power BI

Johan Decorte
February 2024

Course materials: <https://github.com/jdecorte/powerbi>

Introduction to PowerBI.....	2
Getting your software & data.....	2
Import vs. Direct Query	4
Star schema, cubes, slicing & dicing,	4
Star schema.....	4
OLAP Cube	5
OLAP Operators.....	5
Basic tables and charts.....	6
Geographical representation	14
Calculated fields.....	14
Publishing reports.....	16
Creating a dashboard.....	17
Measures.....	17
Advanced Calculations.....	19
Power Query.....	20
PDF's.....	21

Introduction to PowerBI

Microsoft started in 2013-2014 developing new tools for reporting, having some unique characteristics like user friendliness, cloud integration and integration of all kinds of data sources. After a rather chaotic startup phase, in which several tools were introduced and withdrawn (e.g. PowerMap) it looks like at the time of this writing the dust has settled and the PowerBI environment has become stable.

PowerBI currently consists of three applications:

- Power BI Desktop: a desktop application to create reports and charts from all kinds of data sources (Excel, Facebook, Google Analytics, SQL Server, Oracle and many, many more) .
- Power BI Service: a cloud solution, integrated in Microsoft Azure where reports are published and viewed.
- Power BI Mobile: a mobile application for viewing reports.

In this course we mainly use the free Power BI Desktop application, which you can install from <https://www.microsoft.com/nl-be/download/details.aspx?id=45331>. You can also install the Power BI app to enjoy the reports you created on your smartphone.

Getting your software & data

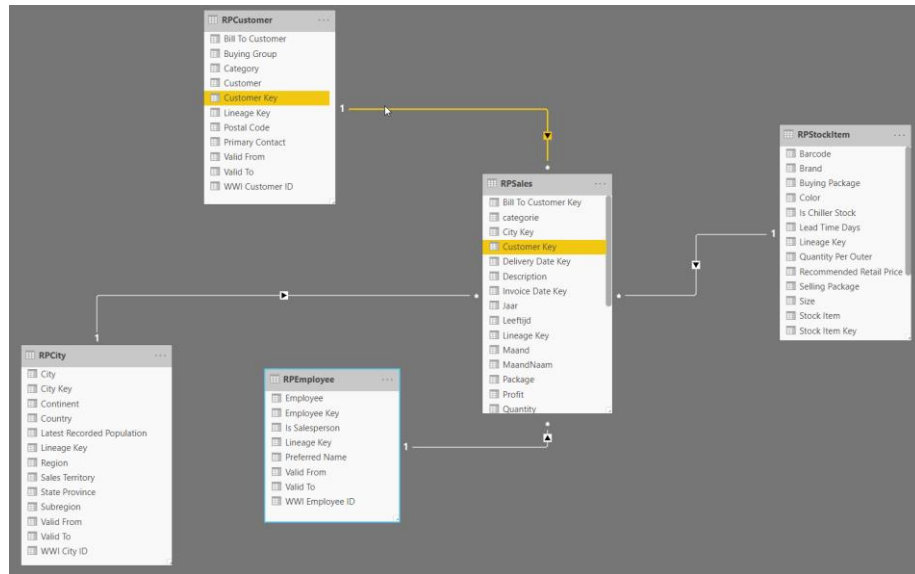
Download and install PowerBI desktop from <https://powerbi.microsoft.com/en-us/desktop/>

1. In case you make your reports from a SQL Server Database

The WorldWideImporters Database is an OLTP system that is mainly used for keeping track of purchases and sales of a retail company. The corresponding WideWorldImportersDW database is also available

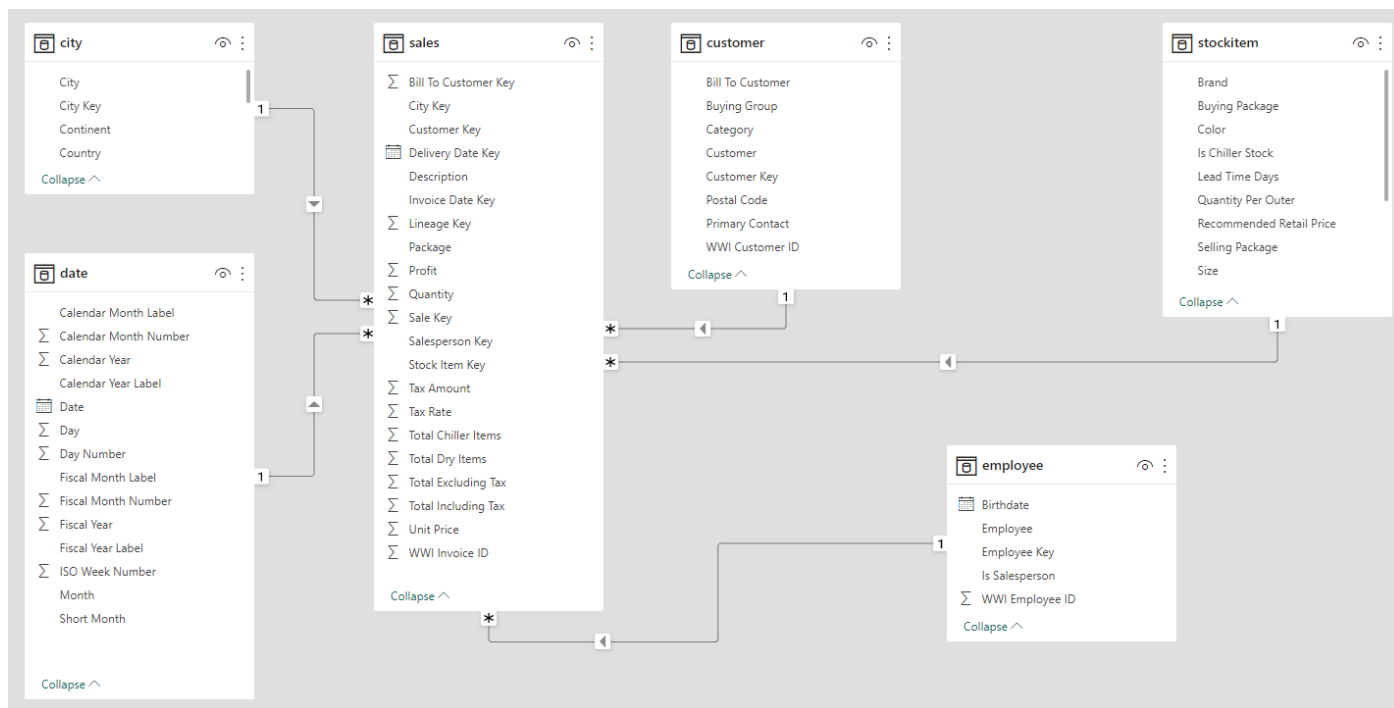
Download and install:

- The OLTP database at: <https://github.com/Microsoft/sql-server-samples/releases/download/wide-world-importers-v1.0/WideWorldImporters-Full.bak>
 - The corresponding OLAP database at: <https://github.com/Microsoft/sql-server-samples/releases/download/wide-world-importers-v1.0/WideWorldImportersDW-Full.bak>
1. Create a connection with your SQL Server Server and OLAP database. Choose to import the tables.
See <https://docs.microsoft.com/en-us/power-bi/service-gateway-sql-tutorial>
 2. Create a data model for the imported tables. Connect the corresponding fields.



2. In case you make your reports from csv and Excel files import following files:
 - a. City.csv
 - b. Customer.csv
 - c. Date.csv
 - d. Employee.csv
 - e. Stockitem.csv
 - f. Sales.csv

Next you have to create the relationships between the tables manually according to the corresponding keys:



Import vs. Direct Query

When referring to a data source (by “Get data”) you can, for most sources, choose for either “Import” or “Direct Query”. If you choose for import then your data is loaded into an in-memory engine (xVelocity, which Microsoft acquired from Vertipaq) where data is structured as a column store database. This allows for very quick aggregation queries but of course feasibility is limited by the available memory on the machine where Power BI runs.

In case of Direct Query you query the data source directly. Power BI queries are translated into the query language of the source (typically SQL in case of a RDBMS). This ensures you always have the actual data but performance is limited by the source system: indexing, load, etc. Also, some functionality is not available in DirectQuery, for example the data button:



Import



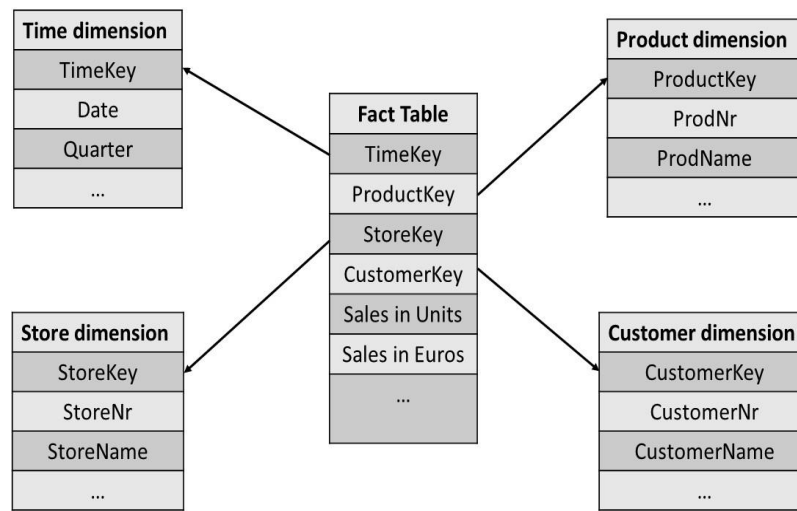
DirectQuery

See <https://docs.microsoft.com/en-us/power-bi/desktop-directquery-about> for more info about DirectQuery.

Star schema, cubes, slicing & dicing, ...

BI tools work best if your data is structured according to the star schema, but, technically, they also work on normalized OLTP databases. We review some topics in this area.

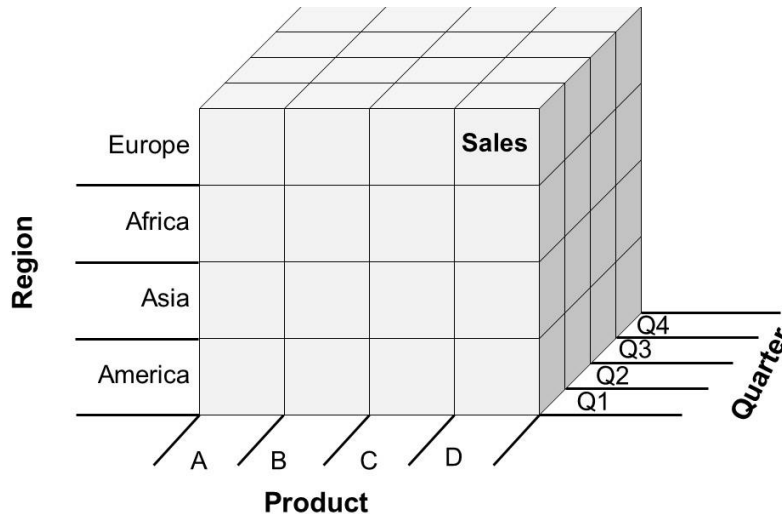
Star schema



- Fact table
 - one tuple per transaction or event (i.e., a fact) and also measurement data (e.g. sales)
- Dimension table
 - further information about each of the facts in the fact table (e.g., Time, Store, Customer, Product).
 - often denormalized = redundancy

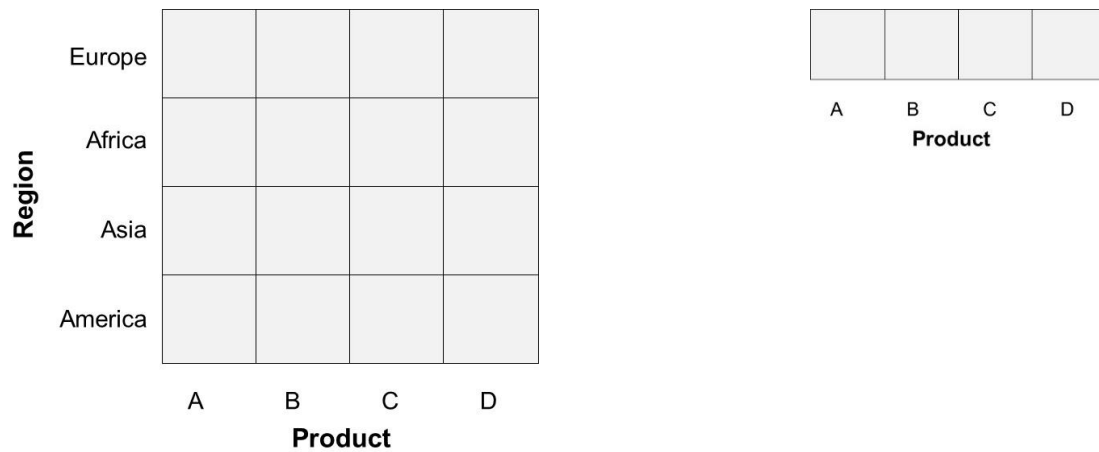
OLAP Cube

Data that is organized in a star schema can be considered as a cube. The dimensions of the star schema are the ribs of the cube. In each dice of the cube you find the values of measures belonging to the fact table. Of course, the dimensions of the "cube" can also be 2, 4, 5, ...

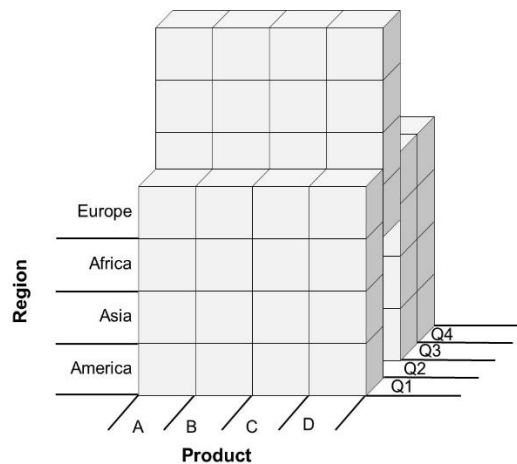


OLAP Operators

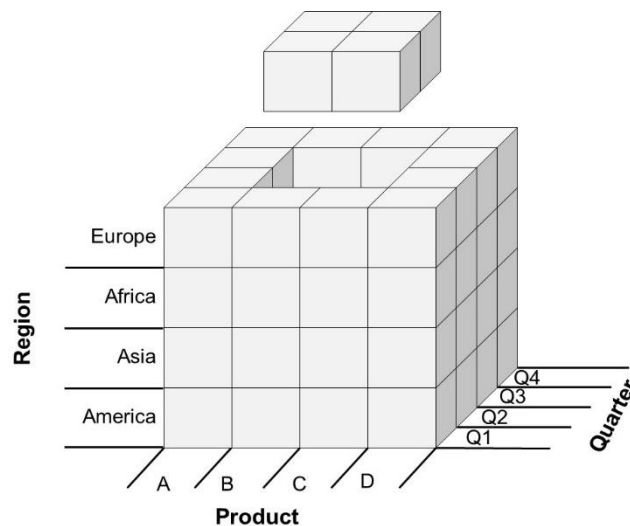
- **Roll-up** refers to aggregating the current set of fact values within or across one or more dimensions. The reverse of roll-up is **drill-down**.



- **Slicing** represents the operation whereby one of the dimensions is set at a particular value.



- **Dicing** corresponds to a range selection on one or more dimensions.



Basic tables and charts


3. Create a simple table to show customer and total number of invoices for that customer for the top 10 customers.

- Create a table using the matrix visual as below. Show "Total including tax" in the rows. Apply conditional formatting (under "Field Formatting" to assign a color to each value. Use filtering to show only the years 2018 and 2019.

Sales Territory	2018	2019	Total
Midwest	1.740M	1.506M	3.246M
Southwest	1.553M	1.391M	2.944M
Texas	747M	713M	1.459M
Rockwall	31M	33M	64M
Oak Point	22M	20M	42M
Dickworsham	21M	18M	39M
Dawn	22M	17M	39M
Jones Creek	21M	17M	39M
Hollywood Park	15M	21M	35M
Haltom City	18M	17M	35M
Kopperl	15M	20M	35M
Marfa	15M	19M	34M
Lytle	22M	12M	34M
Del Valle	16M	17M	34M
Van Alstyne	17M	16M	33M
El Refugio	20M	12M	33M
Universal City	20M	12M	32M
Ben Arnold	16M	16M	32M
Maypearl	15M	17M	32M
The Colony	20M	12M	32M
East Mountain	21M	11M	32M
Impact	15M	17M	32M
Ovilla	18M	13M	32M
Morita	15M	17M	31M
Dorchester	15M	17M	31M
Flowella	17M	14M	31M
Coupland	16M	15M	31M
Buchanan Lake Village	15M	15M	30M
Helotes	14M	16M	30M
Total	11.688M	10.316M	22.003M

What kind of table is created by the matrix visual?

Observe what you can do with the Territory Hierarchy.

- Above we used the Date Dimension table that we imported, which comes in handy if you need calendar years, month, etc. in you report. In case you don't have such a table you can ask to create the required date dimension table for you. Go to the Table view (). In the menu Table tools choose New Table and in de fill-in field type:

Dimension Date = CALENDAR(DATE(YEAR(TODAY())-9,1,1),TODAY())

CALENDAR(*date1,date2*) is a DAX function that creates a table with all dates in the given range, in this case for the last 10 years. Alternatively, you could also write

Dimension Date = CALENDARAUTO()

CALENDERAUTO() creates a table with all dates between the 1st of January of the oldest year in the data and de 31st of December of the last year in the data. However, if your data contains "old" dates (e.g. birth dates) in periods you will never use in reports you will have a very long period in your table. In that case you can better restrict the generated dates to e.g. the period in which you have sales.

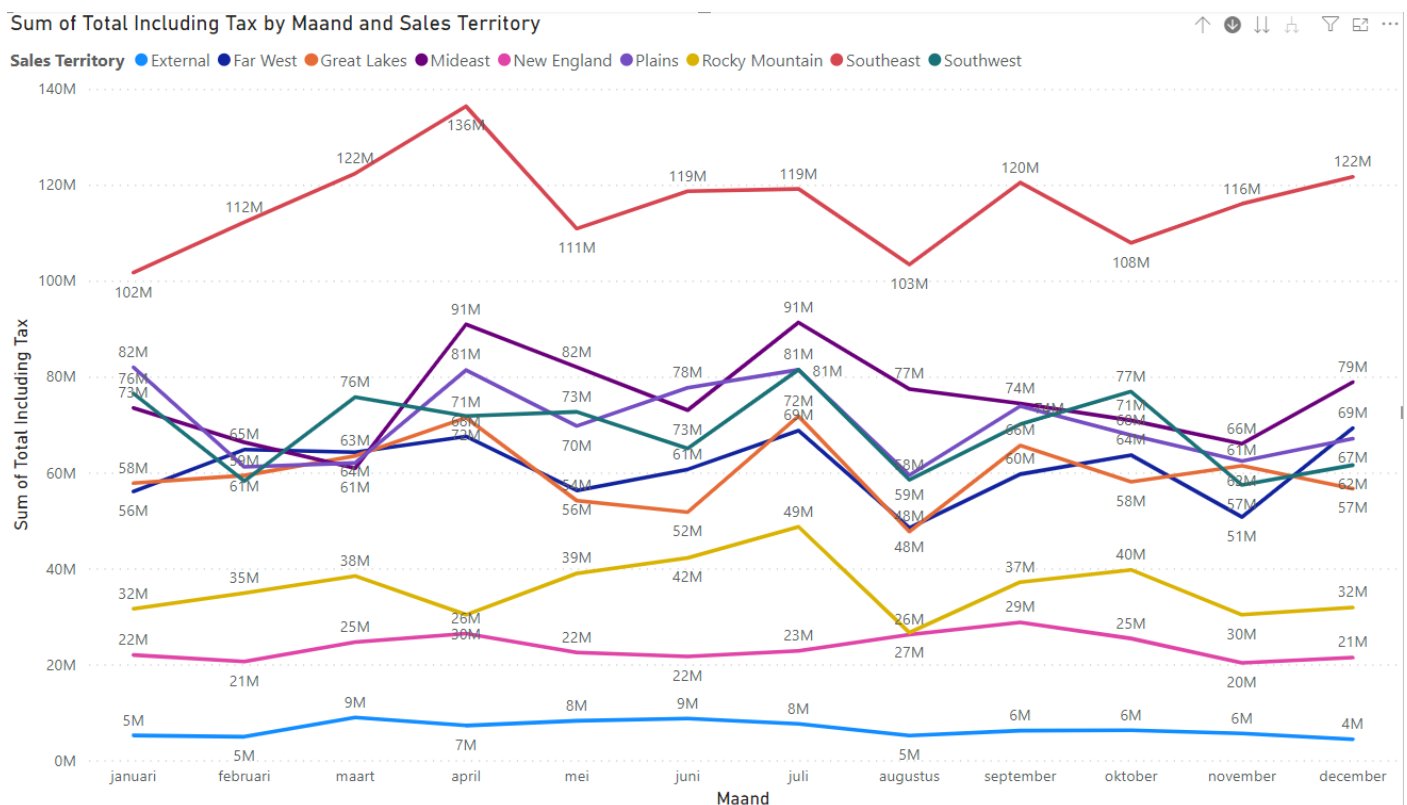
DAX (Data Analysis eXpressions) is a functional programming language that comes with Power BI, which we will gradually introduce during this course.

If needed you can add this table to the data model and create a relation between [Fact Sale].[InvoiceDate] and [Dimension Date].[Date].

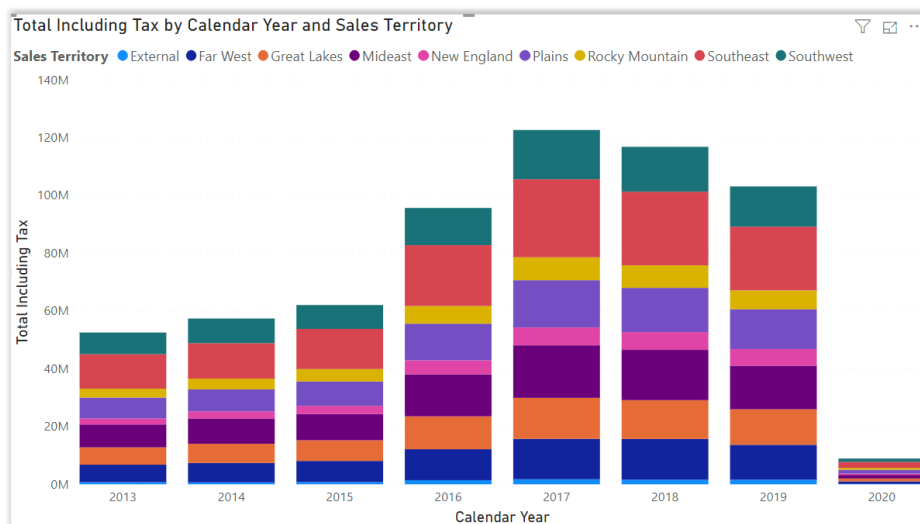
6. Line chart

Observe that for the Date field in the Date table a hierarchy is created automatically.

- Create a line chart that shows the evolution of the [Total Including Tax] for each Sales Territory over time.
- Use drilldown to view the evolution by month.
- Add a slicer for the calendar year.
- Add local minima and maxima.

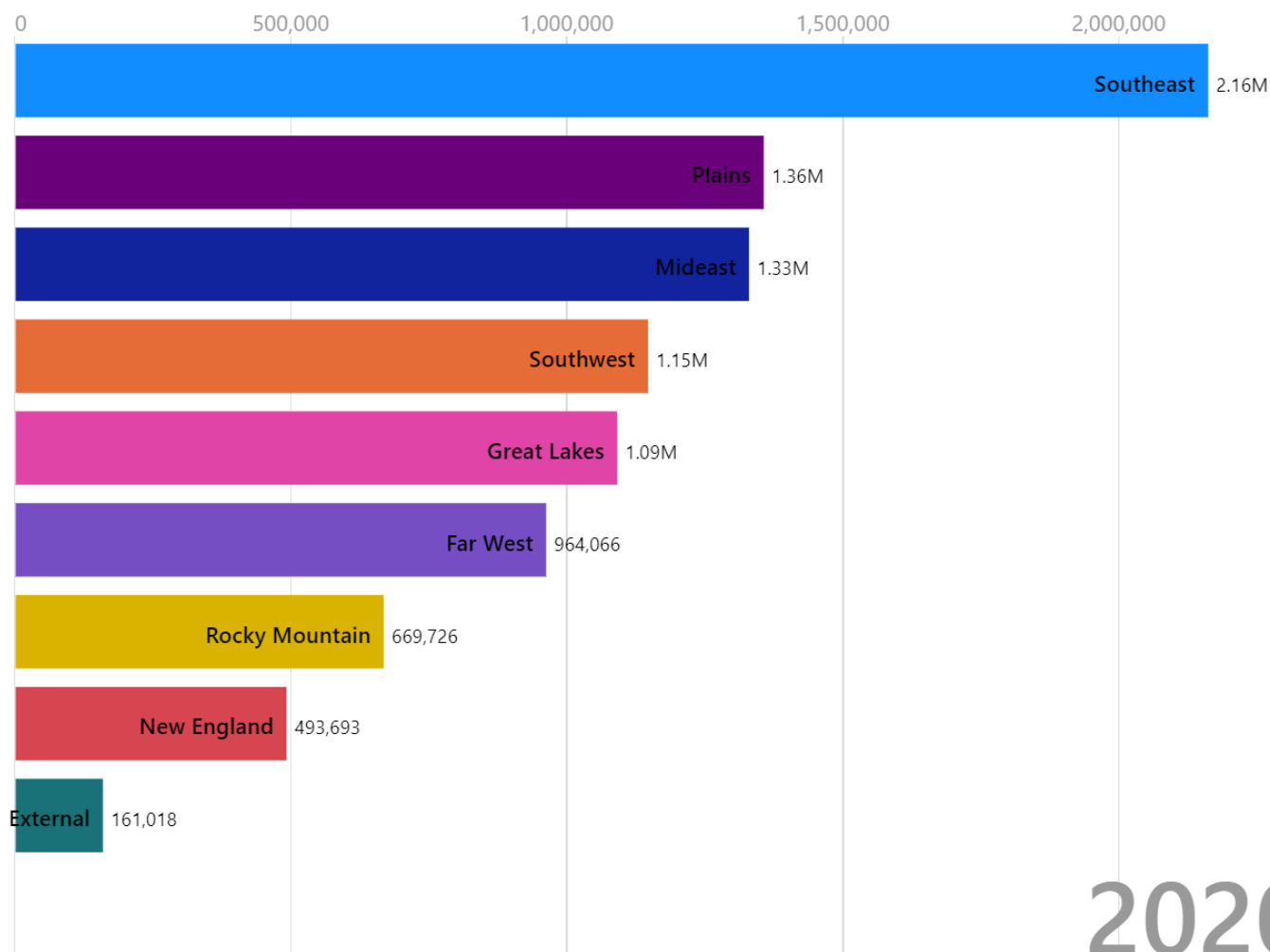


7. On the next page add a stacked column chart as below.



8. Power BI contains lots of visuals provided by the community. Click “Get more visuals” (...) and search for “Animated Bar Chart Race”. Click “Add”. Turn the above bar chart into an animated bar chart race:

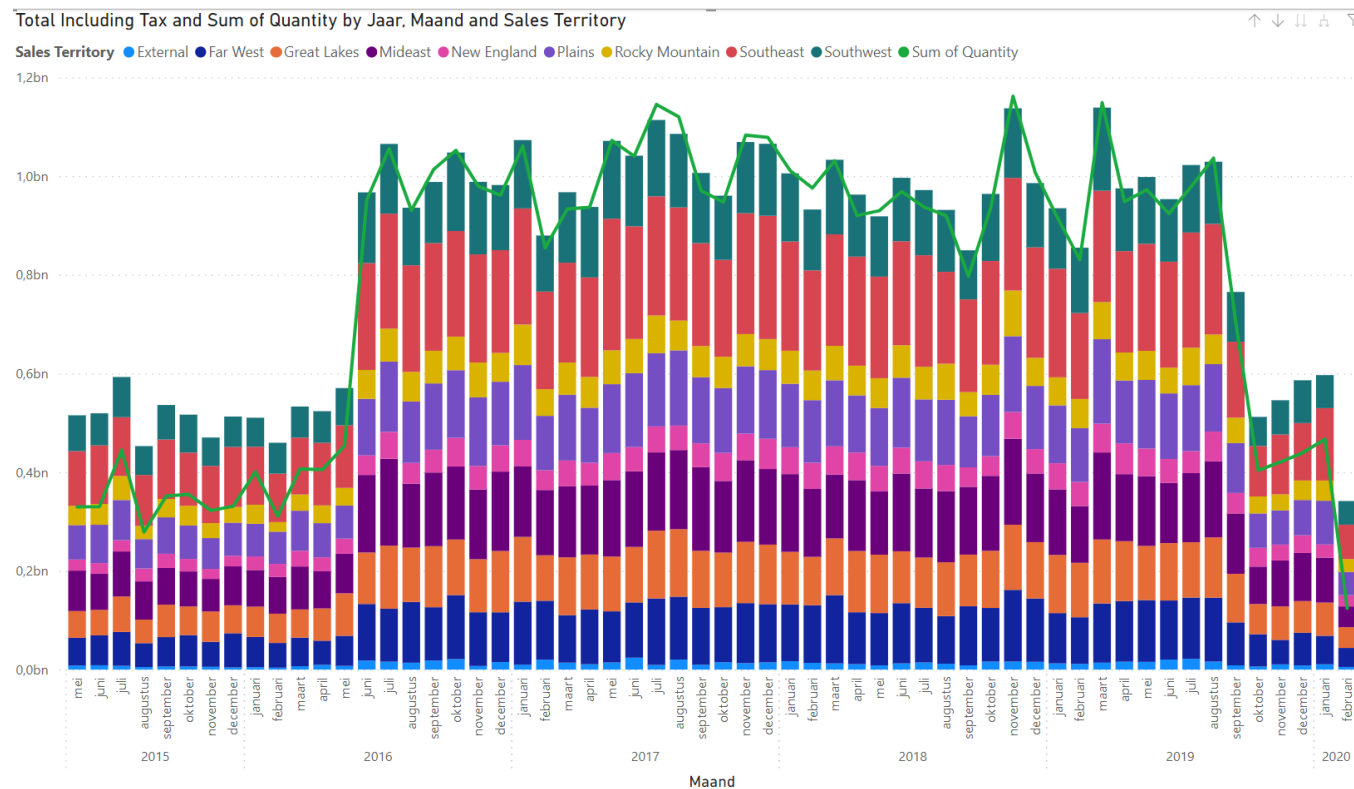
Sum of Total Including Tax by Sales Territory and Calendar Year



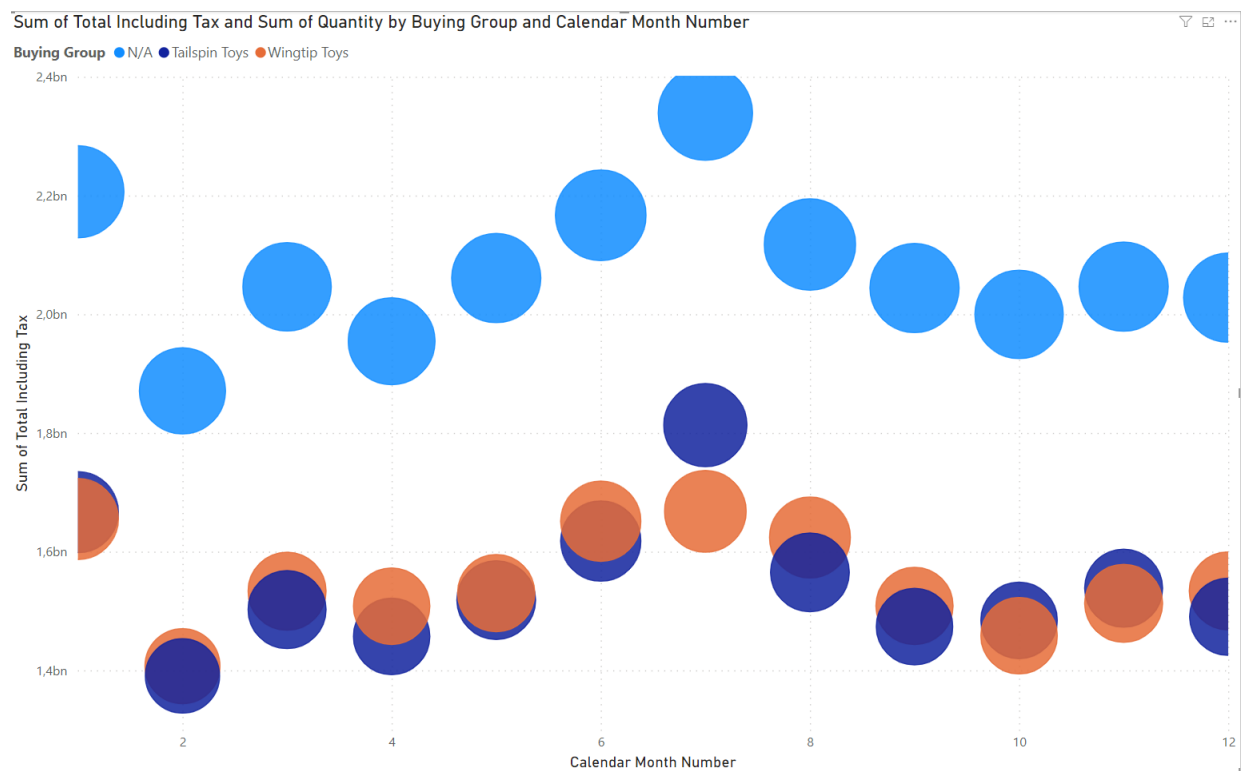
Also install the visual “Play axis” to control play and pause. Make sure both visualizations have the same “Time” setting.

9. Stacked / Line chart

A stacked/line chart allows to combine (and compare) different (but related) values, e.g. Total Including Tax and Quantity, on two different axes.

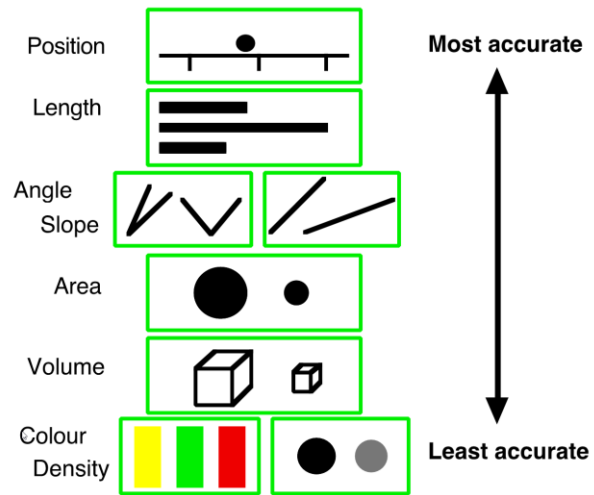


10. Apply the slicing operator by adding two list based slicers to the page: one for the calendar years and one for the Employee? Make sure only Salespersons are shown in this last slider by adding a filter. Explore what happens if you click the slicers. Which two OLAP operators are illustrated here?
11. Add a bubble chart (use the Scatter chart icon) to illustrate the relation between four variables: Calendar Month Number, Total Including Tax, Quantity and Buying Group. See example below.



12. How can we improve the visualization of the data in the previous chart given the results of research below.

Accuracy of judgment of encoded quantitative data



Cleveland and McGill (1984)

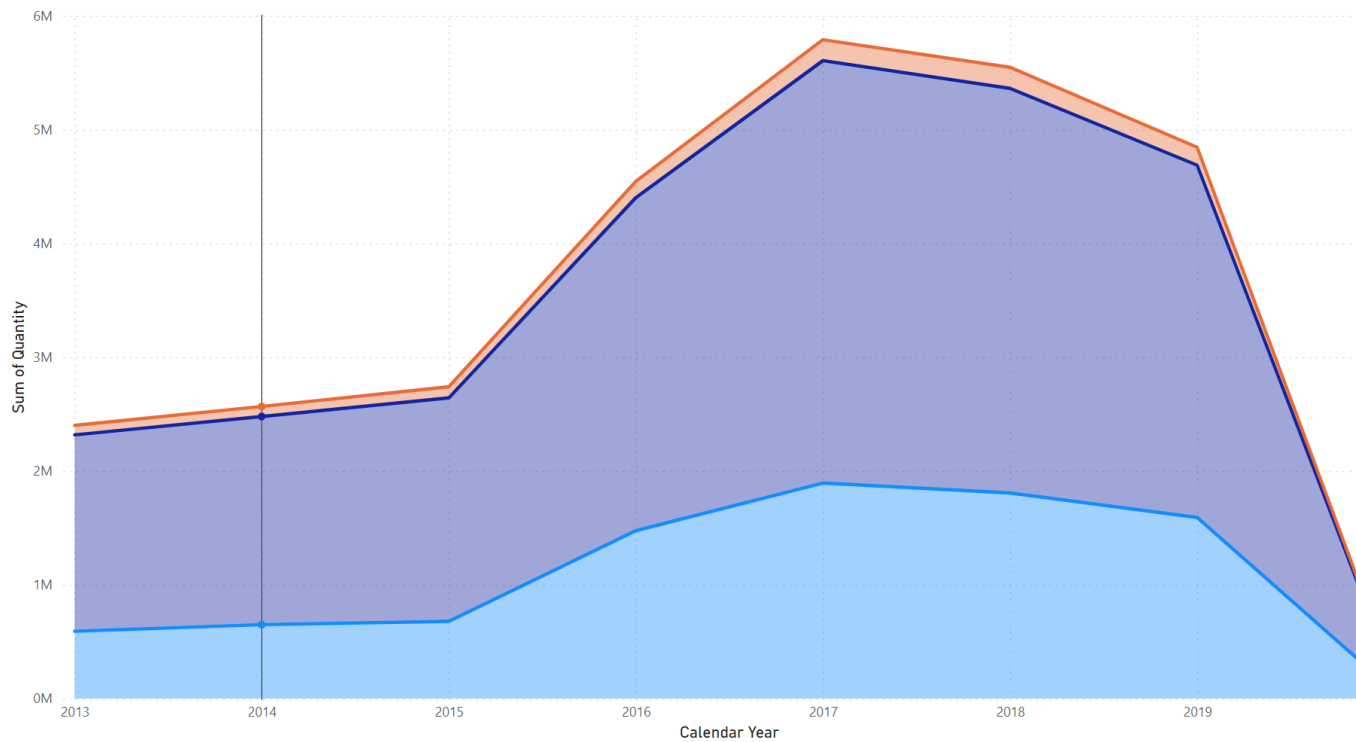
13. Create a treemap with drill-drill from Territories to Cities as Category and Quantity as value.



14. Draw a stacked area chart showing the evolution of the sold quantities by year and by buying package.

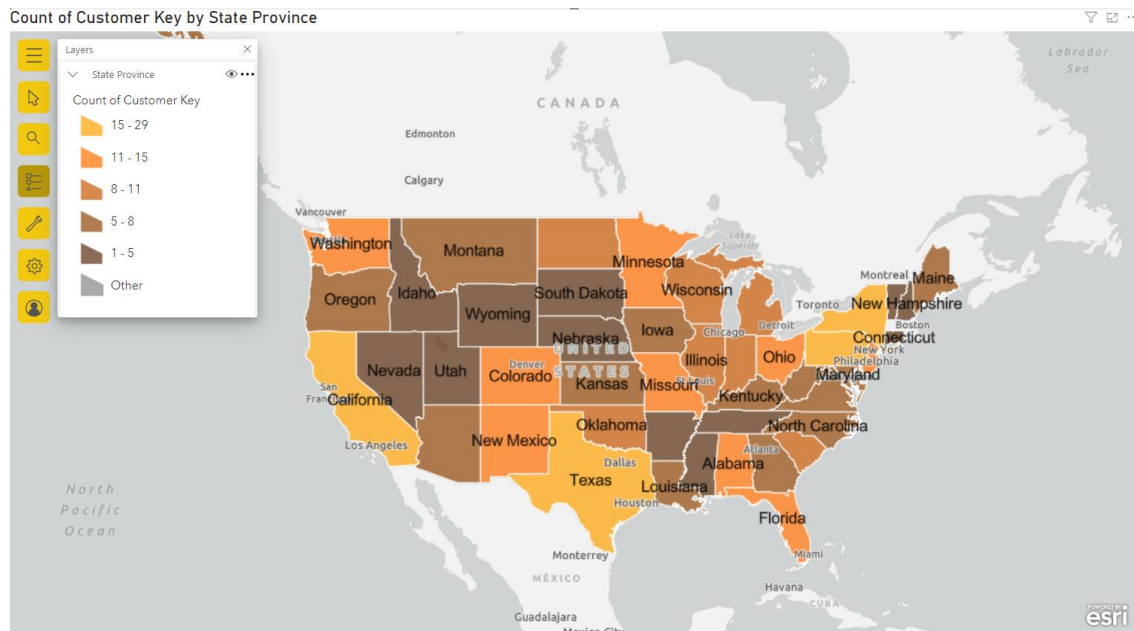
Sum of Quantity by Calendar Year and Buying Package

Buying Package ● Carton ● Each ● Packet



Geographical representation

15. To introduce the capabilities of the third party ArcGIS visual we can use the field State Province to visualize the number of real customers per state (i.e. the number of customers who actually bought something). Drag and drop the field State Province to the Location and the Customer Key (from table sales) to the Color. In the dropdown of color select "Count(Distinct)".

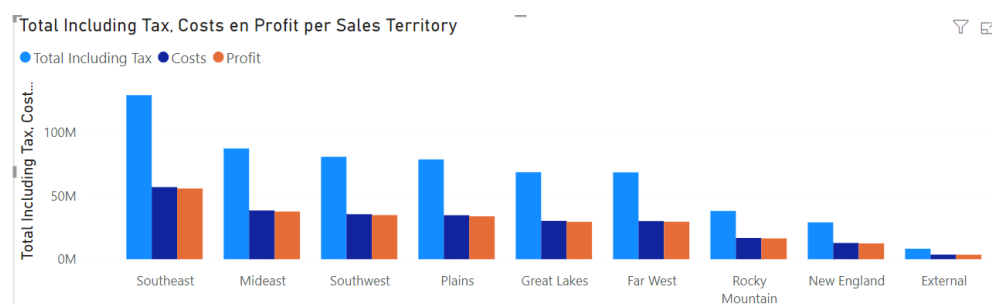


Calculated fields

16. We have the sales and the profit but not the cost. If we want to visualize the cost we can add a "new column" to calculate the cost by clicking the ... next to the table naming and writing the formula.

$$\text{Costs} = [\text{Total Excluding Tax}] - [\text{Profit}]$$

We can now draw e.g. a "Clustered Column Chart":

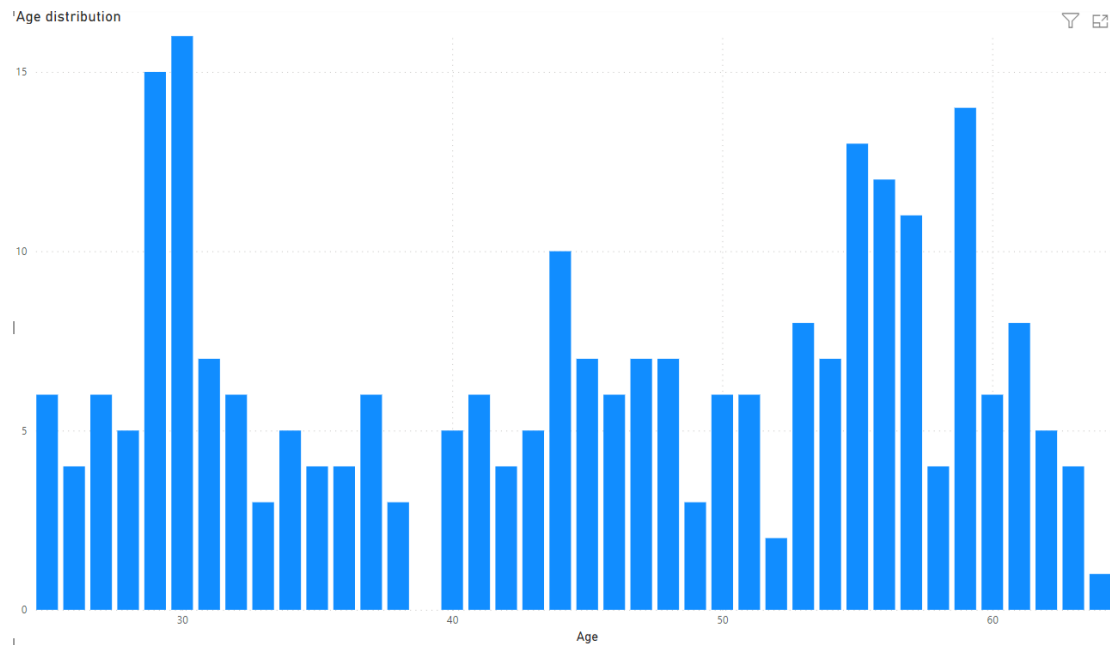


17. Numerous built-in DAX functions can be used in your formulas.

To calculate the age of a person, supposing you have the birthdate, you can add a calculated column to the Employee table with the formula below:

$$\text{Age} = \text{DATEDIFF}([\text{Birthdate}], \text{TODAY()}, \text{year}) - \text{IF}(\text{MONTH}([\text{Birthdate}]) * 100 + \text{DAY}([\text{Birthdate}]) > \text{MONTH}(\text{TODAY()})*100 + \text{DAY}(\text{TODAY()}), 1, 0)$$

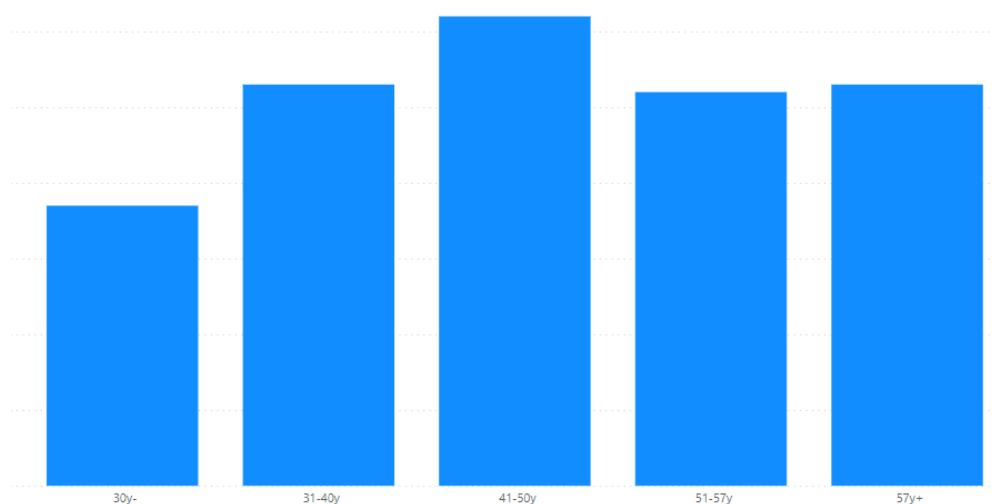
Then you can use this column to create an age distribution diagram:



18. Create distribution diagram for age categories

Add yet another calculated column with the formula

```
Age Category = if([Age] <= 30,"30y-",if([Age] <= 40,"31-40y",if([Age]<= 50,"41-50y",if([Age]
<= 57,"51-57y", "57y+"))))
```



Remember to sort ascending by Age Category.

Some other interesting calculation are:

- Left([column],number): left *number* of characters from [column].
- Right([column], number): right *number* of characters from [column].
- Mid([column],start, number): give *number* of characters from [column] as of *start*
- Year([datecolumn]), Month([datecolumn]), Day([datecolumn]): extract year, month, day from a data. Pay attention: this only works if the column is considered as a date type by PowerBI, not if it is a string or number. You can change the data type in the modeling menu.
- Formatting a date, e.g.

Format([datecolumn],"YYYY-MMM-DD") → 2023-apr-05.

See <https://docs.microsoft.com/en-us/dax/custom-date-and-time-formats-for-the-format-function> for a complete list of all available formats.

For a complete list of all functions, see <https://docs.microsoft.com/en-us/dax/dax-function-reference>

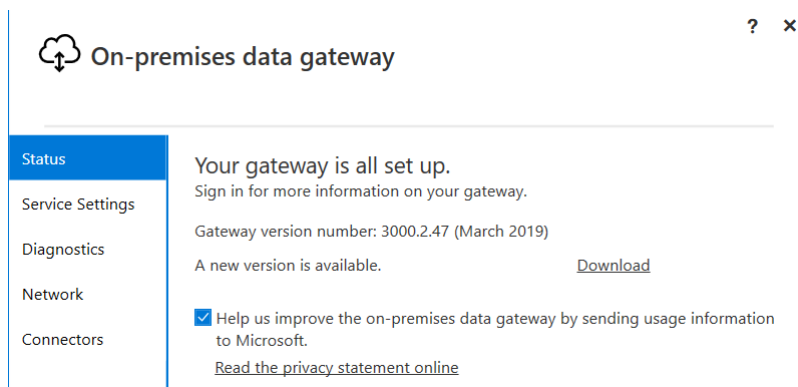
Publishing reports

Create a PowerBI account (you can start with the free version via “Use it free”) on <http://app.powerbi.com>. Important remark: you need a professional e-mail address to be able to create an account. E-mail addresses @hotmail.com, @gmail.com, @telenet.be, etc. won't work. So USE YOUR WORK OR SCHOOL E-MAIL ADDRESS.

Now you can easily publish reports and view them in a browser using PowerBI service. If your reports are of type “Import” the data is published to the online service (on Azure) and only updated if you manually republish.

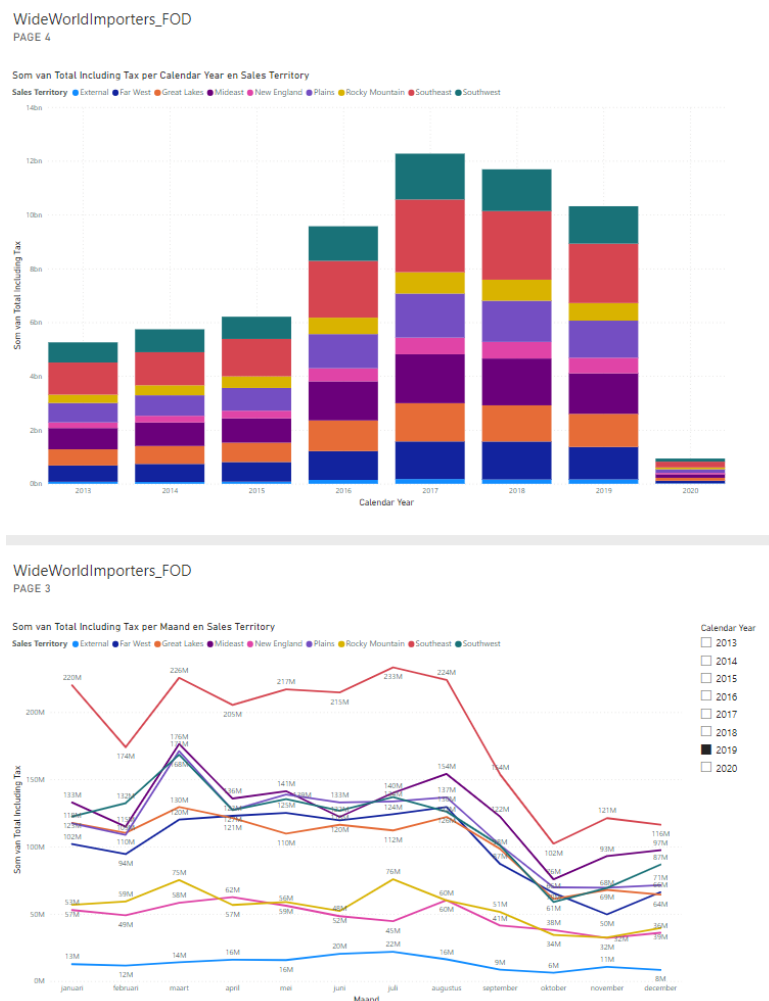
One way to always see the actual data in the service is by using Excel or csv files you store in OneDrive, the file share (say Dropbox like) solution of Microsoft. As a Office365 user you get 1TB online storage. If you are not an Office365 user you can use OneDrive Personal. In this use case you have to download your query results from your database using a scheduler (that runs the query and extracts the results to a file e.g. every hour). Power BI service guarantees that it looks at least every 30 minutes for updates in the OneDrive files.

Another way to always get access to the real time data is by using Direct Query when building your report. However, then the Power BI service needs access to your on-premise database which can be accomplished by installing the Power BI on-premise gateway service on your local server. Of course, Microsoft pushes you to use SQL Database on Azure, which solves all these problems.



Creating a dashboard

Once you have published your report to the Power BI Service you can easily add it to a dashboard by going to the online report and clicking “Pin to dashboard” (click ... first).



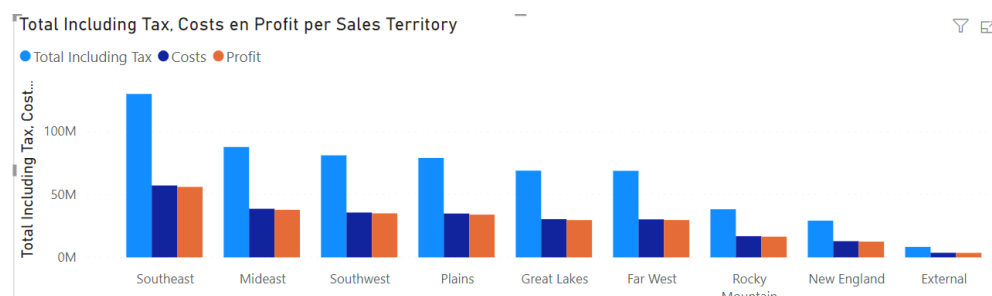
Slicers can also be published to a dashboard as part of a report, so dashboards are also interactive.

Measures

- If you want to calculate e.g. the costs, it's ok to create a new column, see above.

Costs = [Total Excluding Tax] - [Profit]

The costs will correctly aggregate over e.g. [Sales Territory], by whatever filter:



However, if you want to calculate the Profit *as a percentage* of the Total Excluding Tax you can't simply add a new column as [Profit]*100/[Total Excluding Tax] as percentages do not aggregate correctly: it does not make sense to make a sum or average of percentages. To cope with this you need a Measure:

ProfitPct = sum([Profit])*100/sum([Total Excluding Tax])

A measure is indicated by the icon of a calculator.

Aggregation and totals (see below) now are correctly calculated, no matter which filtering or aggregation is used:

Employee	ProfitPct	Calendar Y...
Amy Trefl	49,21	<input type="checkbox"/> 2013
Anthony Grosse	49,52	<input type="checkbox"/> 2014
Archer Lamble	49,19	<input type="checkbox"/> 2015
Hudson Hollinworth	49,14	<input type="checkbox"/> 2016
Hudson Onslow	49,15	<input checked="" type="checkbox"/> 2017
Jack Potter	49,44	<input type="checkbox"/> 2018
Kayla Woodcock	49,54	<input type="checkbox"/> 2019
Lily Code	49,63	<input type="checkbox"/> 2020
Sophia Hinton	49,16	
Taj Shand	48,97	
Total	49,30	

Often used DAX functions for measures are SUM(), AVERAGE(), MIN(), MAX(), COUNT(), COUNTROWS() and DISTINCTCOUNT(). The latter function counts the number of unique values in a column. COUNT() counts the number of times the field it gets as a parameter is not blank, COUNTROWS() counts the number of rows in a table.

Unlike a calculated column a measure is not a new column that is physically added to a table. Instead all its values are recalculated for each aggregation and filter that is applied. As such a measure can't be provided in the data warehouse.

20. We now make a Multirow Card with following data: number of distinct customers, maximum of [Total Excluding Tax], number of invoice lines, highest profit amount and lowest profit amount. We have to link following measures to the table Fact Sale:

1. Distinct Customers = DISTINCTCOUNT([Customer Key])
2. MaxSales = MAX([Total Excluding Tax])
3. InvoiceLines = COUNTROWS('sales')
4. MaxProfit = MAX([Profit])
5. MinProfit = MIN([Profit])

Finally, adding slicers for State/Province and Year illustrates that all values change accordingly.

32 Distinct Customers	9.200,00 MaxProfit	-452 MinProfit	43709 InvoiceLines	18.990,00 MaxSales					
Select all	Alabama	Alaska	Arizona	Arkansas	California	Colorado	Connecticut	Delaware	District of...
Florida	Georgia	Hawaii	Idaho	Illinois	Indiana	Iowa	Kansas	Kentucky	Louisiana
Maine	Maryland	Massachuset...	Michigan	Minnesota	Mississippi	Missouri	Montana	N/A	Nebraska
Nevada	New...	New Jersey	New Mexico	New York	North Carolina	North Dakota	Ohio	Oklahoma	Oregon
Pennsylvania	Puerto Rico...	Rhode Island	South...	South Dakota	Tennessee	Texas	Utah	Vermont	Virgin Islan...
Virginia	Washington	West Virginia	Wisconsin	Wyoming					

2013	2014	2015
2016	2017	2018
2019	2020	

Advanced Calculations

21. Suppose you want to calculate the number of invoice lines and the number of invoices per customer. You could do this in a simple table visual, but it requires a lot of configuration on Visual level and it is not reusable in other Visuals. Therefore it's often better to use measures instead. The DAX function `RELATEDTABLE(table)` returns the rows from the table you pass as a parameter that are related to the row of the calculated field. Of course, you can't have a complete table as the result of a calculated field, so you have to do a calculation with that table. For the number of `invoicelines` we can add a calculated column to the customer table as follows:

```
NrOfInvoiceLines = COUNTROWS(RELATEDTABLE('sales'))
```

To find the number of invoices per customer you have to count all distinct [WWI Invoice ID] values for each customer in the sales table. This can best be accomplished by adding a measure to the sales table:

```
NrOfInvoices = DISTINCTCOUNT([WWI Invoice ID])
```

Now it's easy to make the requested table list (or any other visual with these measures):

Customer	NrOfInvoiceLines	NrOfInvoices
Tailspin Toys (Absecon, NJ)	1108	361
Tailspin Toys (Aceitunas, PR)	1041	326
Tailspin Toys (Airport Drive, MO)	1060	338
Tailspin Toys (Alstead, NH)	1057	325
Tailspin Toys (Amanda Park, WA)	1049	318
Tailspin Toys (Andrix, CO)	985	316
Tailspin Toys (Annamoriah, WV)	1058	327
Tailspin Toys (Antares, AZ)	1067	329
Tailspin Toys (Antonito, CO)	1071	330
Tailspin Toys (Arbor Vitae, WI)	984	317
Tailspin Toys (Arietta, NY)	1077	337
Tailspin Toys (Armstrong Creek, WI)	1111	351

Power Query

22. Power Query can be considered as a kind of ETL tool because it allows you to create a script for cleaning, transforming and aggregating data before you make your reports. We'll illustrate this by loading a csv file biomethane.xlsx. First explore the file in Excel.

Do not load the file but click "Transform Data".

Click "use first row as headers".

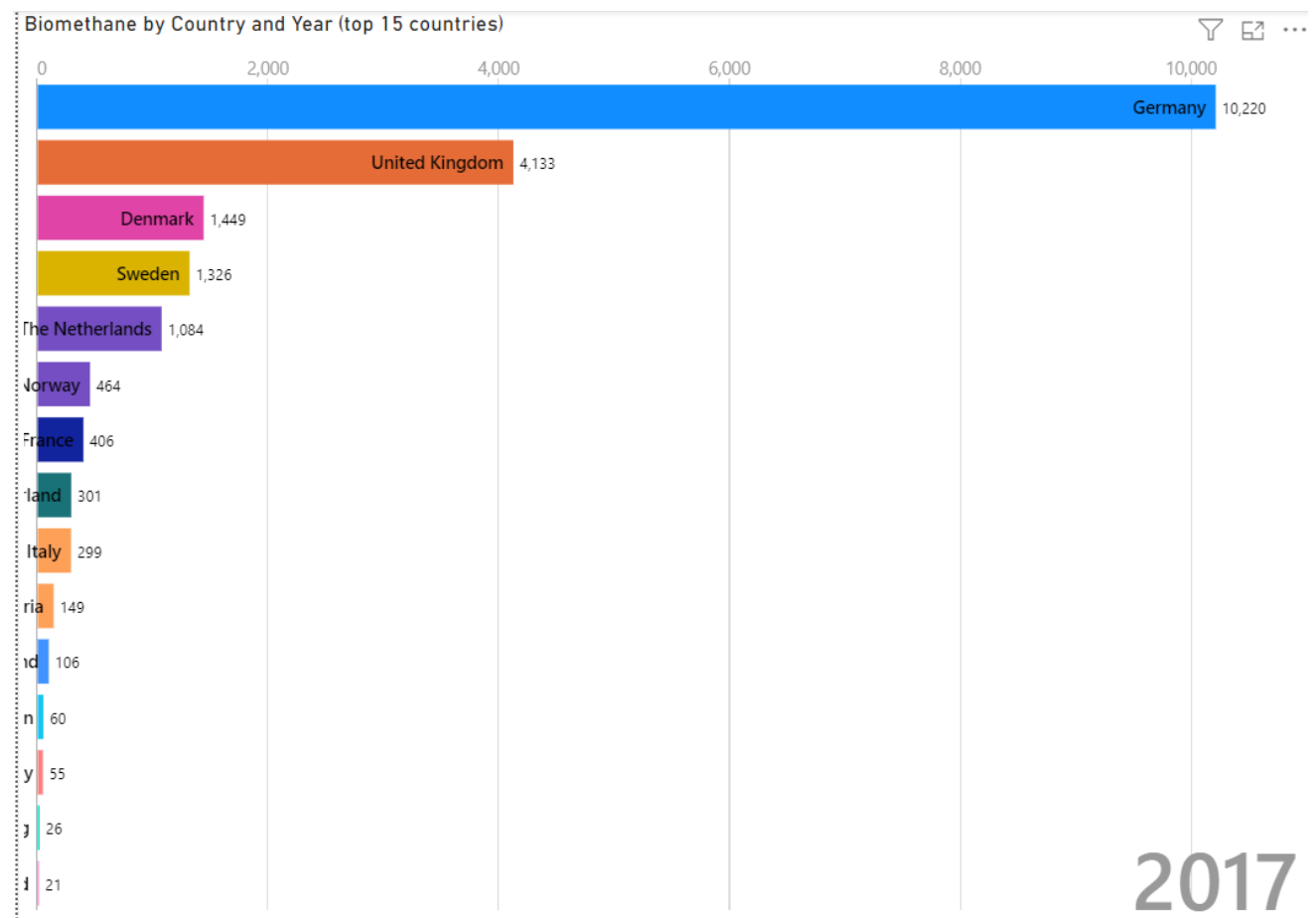
In the transform menu "unpivot" (NL: "Draaitabel opheffen voor kolommen") the columns. Select the column "Country" and choose "Unpivot other columns".

Rename the column attribute to year.

Click "Close and apply" in the "Start" menu.

Exercise:

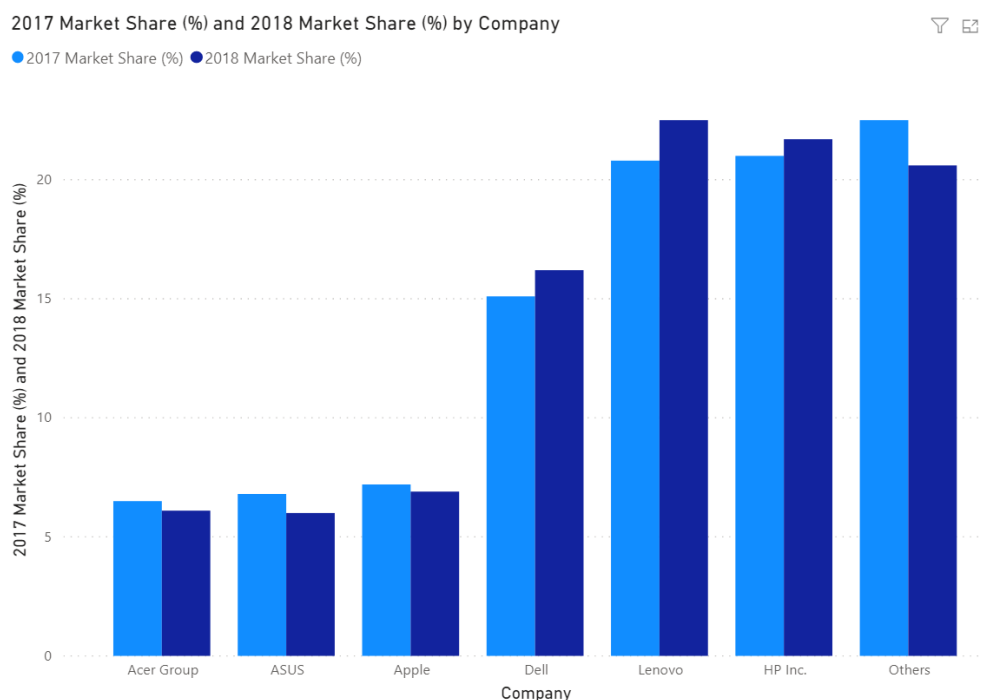
Create an animated bar chart to show the evolution of the biomethane production of each country over the years:



PDF's

23. PowerBI can also identify tables in PDF documents. For this exercise we use a Gartner article about US and worldwide PC sales available at <https://www.gartner.com/en/newsroom/press-releases/2019-01-10-gartner-says-worldwide-pc-shipments-declined-4-3-perc#:~:text=Worldwide%20PC%20shipments%20totalled%2068.6,1.3%20percent%20decline%20from%202017.>

1. First print this website as PDF and save it to your local directory
2. See what happens if you load (transform the data before loading) this pdf file into Power BI. Choose only the table about **Table 3. Preliminary Worldwide PC Vendor Unit Shipment Estimates for 2018 (Thousands of Units)**.
3. What happened when importing the numbers? Why did this happen?
4. In column "2018 Market Share (%)" the decimal . (US notation) has been removed, so we have to divide the percentage by 10 to get the correct numbers. Select that column and choose "Add column" in the menu and in group "From Number" choose Standard/Divide and enter 10. Remove the original column and rename the newly created column to "2018 Market Share (%)".
5. Do the same for 2017.
6. Remove the column "2018-2017 Growth (%)"
7. Remove the Total row (specify to remove the last row)
8. Rename the table to WorldWidePCSales2018
9. To check the data create a clustered column chart that shows the market shares in 2017 and 2018. Order ascending by Company.



Time intelligence

24. DAX has a number of functions for filtering on time. They are called Time Intelligence functions and are often used in measures. Time intelligence functions only work in import modus, not in Direct Query. For the time intelligence functions to work it is also required that for every date from the period in which you have facts (e.g. sales) there is a date in the date dimension, even if there are no sales on that date¹.

We are now going to use these functions together with the `CALCULATE()` function. `CALCULATE()` is a very important function in DAX, but its use is rather complex. It allows you to make calculations with a changed filter context. It takes an expression as its first argument, which is either a measure or an aggregated function, like `SUM()`, `COUNT()`, etc. This means `SUM[Sales]` is equivalent to `CALCULATE(SUM([Sales]))`. However `CALCULATE()` allows you to change the context in which the expression from the first parameter is evaluated by adding extra parameters, which are in fact filters.

Let's start with an example. Businesses often want to compare sales in a year to the previous year. All reports we have seen so far are filtered according to their context. For instance, if a matrix visual has the years as column headers, all values in a column are filtered for the corresponding year. The function `PREVIOUSYEAR()` changes the filter context to dates a year before. Now we create a new measure as follows:

```
SalesYr-1 = CALCULATE(SUM([Total Excluding Tax]),PREVIOUSYEAR('date'[Date]))
```

If we use the measure `SalesYr-1` in a visual we have to make sure that `SalesYr-1` is always defined. Therefore we restrict the table to the years 2016-2019. It also does not make sense to display the last year 2020 because it is not complete.

You can remove the decimals and the column subtotals in the formatting tab.

¹ That's exactly what can be accomplished by using the `CALENDAR()` function to create the `Dimension Date` or by using an imported `Date` table that contains all dates in the period.