



# WorkshopPLUS - Data AI: Business Analytics with Power BI

Module 1



# **CONDITIONS AND TERMS OF USE:**

© Microsoft Corporation. All rights reserved.

You may use these training materials solely for your personal internal reference and non-commercial purposes. You may not distribute, transmit, resell or otherwise make these training materials available to any other person or party without express permission from Microsoft Corporation. URL's or other internet website references in the training materials may change without notice. Unless otherwise noted, any companies, organizations, domain names, e-mail addresses, people, places and events depicted in the training materials are for illustration only and are fictitious. No real association is intended or inferred. THESE TRAINING MATERIALS ARE PROVIDED "AS IS"; MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED IN THESE TRAINING MATERIALS.

# Learning Units covered in this Module

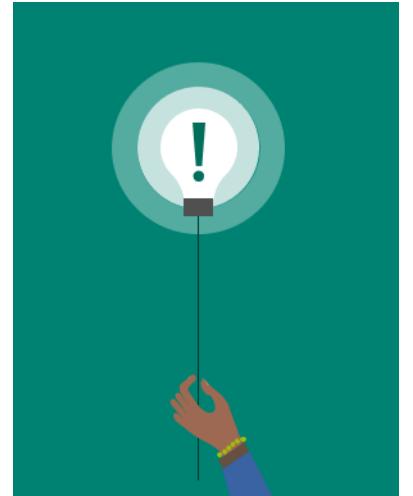
- Lesson 1: Motivation
- Lesson 2: Connecting to Data
- Lesson 3: Shaping Data
- Lesson 4: Creating a Data Model
- Lesson 5: Calculations
- Lesson 6: Building Reports
- Lesson 7: Sharing

# Lesson 1: Motivation

# Objectives

After completing this learning, you will be able to:

- Describe Power BI components and its benefits to business



# Today, BI extends to everyone

**3<sup>rd</sup> wave**  
End user BI



Everyone

**2<sup>nd</sup> wave**  
Self-service BI



Analyst to end user

**1<sup>st</sup> wave**  
Technical BI



IT to end user

# Turning data into business insights is challenging

Common BI challenges include...



## End-to-end view

Data often resides in disparate locations, making it difficult to see a complete picture of your business



## Multiple data sources

Data residing in cloud solutions and on-premise locations is difficult to access and refresh securely



## Right data for the right users at the right time

Different roles have different needs and business users need the latest operational data

# Gartner's 2020 Magic Quadrant

Gartner recognizes Microsoft as a leader in Analytics and BI platforms for 11 consecutive years. Why have the customers chosen Power BI?

- High Value
- Comprehensive analytics solution
- Centralized management
- Global scale
- Security, governance, and compliance

Figure 1. Magic Quadrant for Analytics and Business Intelligence Platforms

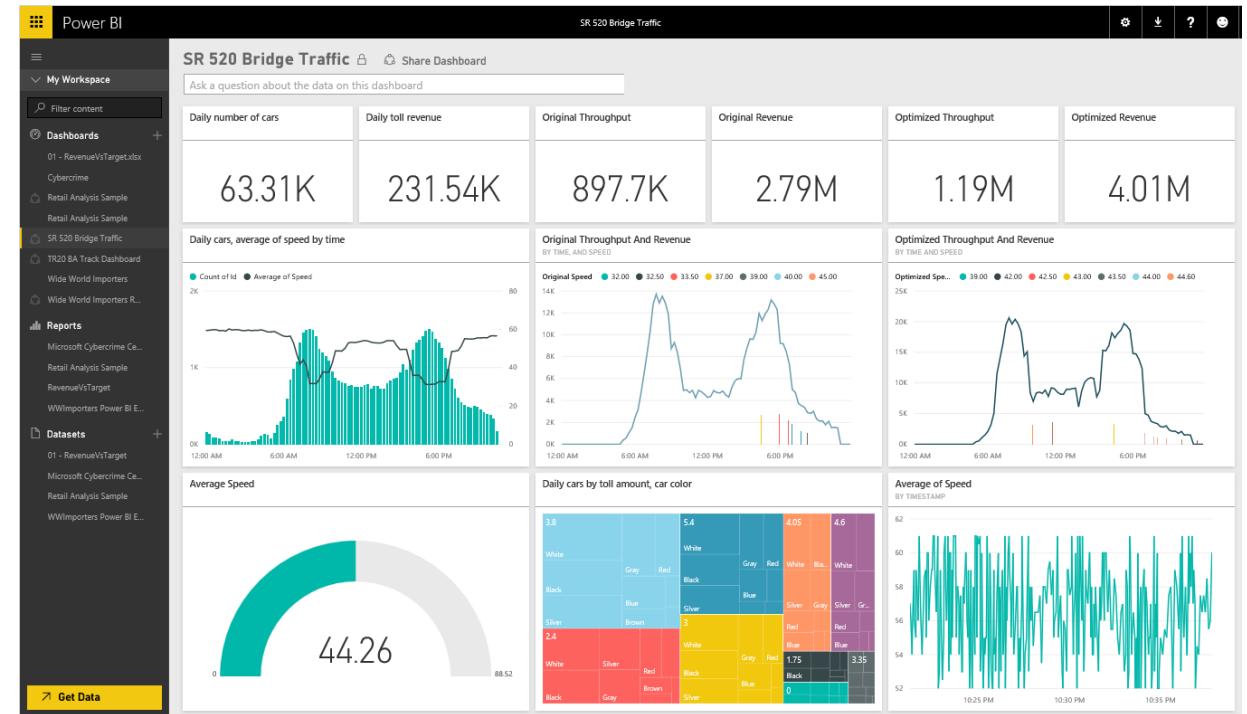


# Power BI

Experience your data. Any data, any way, anywhere

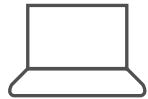
Power BI is a cloud-based business analytics service that enables:

- Fast and easy access to your data
- A live 360° view of your business
- Data discovery and exploration
- Insights from any device
- Collaboration across your organization
- Anyone to visualize and analyze data



# Power BI product portfolio

Author



Power BI  
Desktop

Free data analysis  
and report authoring  
tool

Share and collaborate



Power BI  
service

Cloud-based modern  
business analytics  
solution

Large scale deployments



Power BI  
Premium

Dedicated capacity for  
increased performance

Share and collaborate



Power BI  
Report Server

On-premises report  
server (updated 3x  
per year)

App dev



Power BI  
Embedded

Visual analytics  
embedded in your  
applications

# Power BI Overview



# Create Power BI Content

What is Power BI Desktop ?

A free downloadable  
32/64 bit desktop  
application optimized  
for the Power BI  
Service

Does not depend on  
Microsoft Office

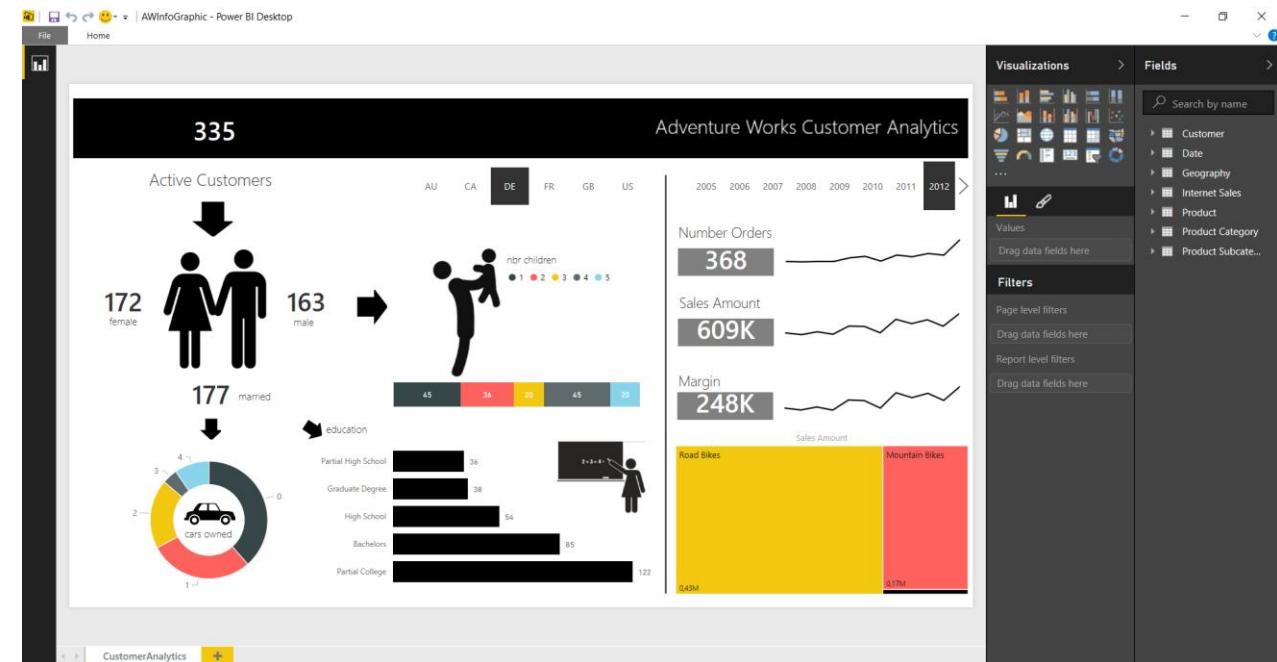
Combines the power  
of Power Query, Power  
Pivot and Power View

Dedicated BI workflow:  
Specializing in  
Interactive  
Visualization and  
Analysis

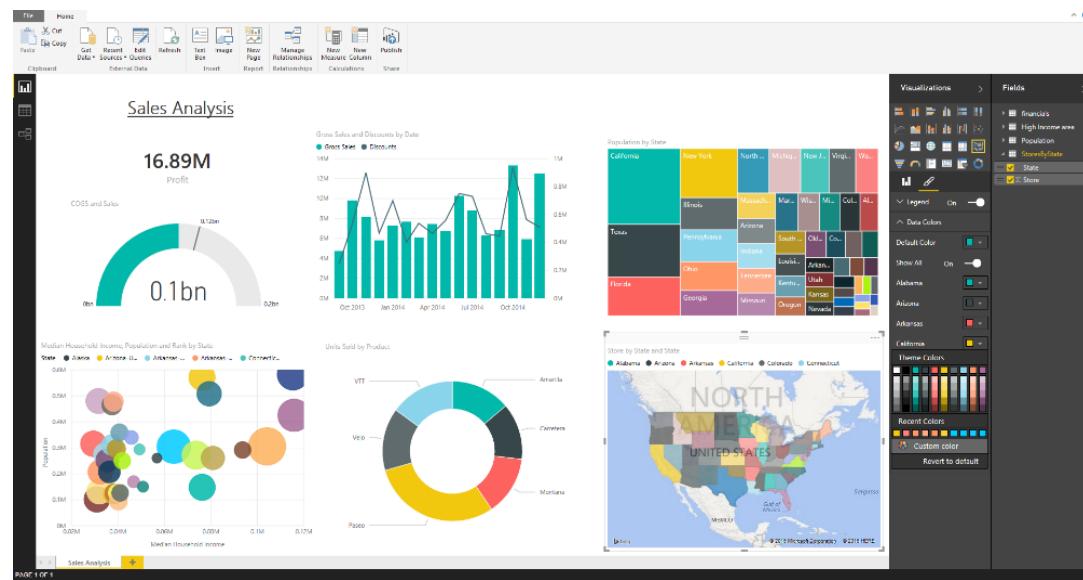
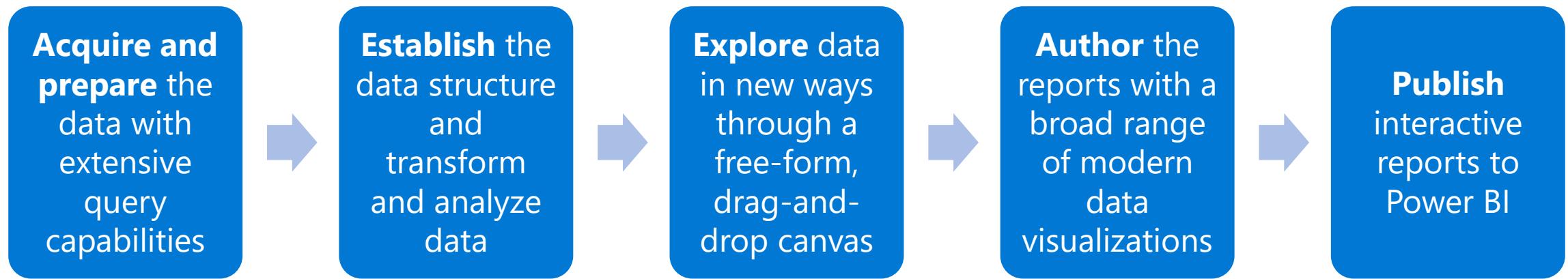
Main output:  
Interactive Reports and  
Data Models

Effortlessly publish  
interactive visual  
reports to Power BI  
users

Support: Monthly  
updates



# Power BI Desktop Workflow



**Questions?**



# Knowledge Check

What are the different business intelligence challenges?

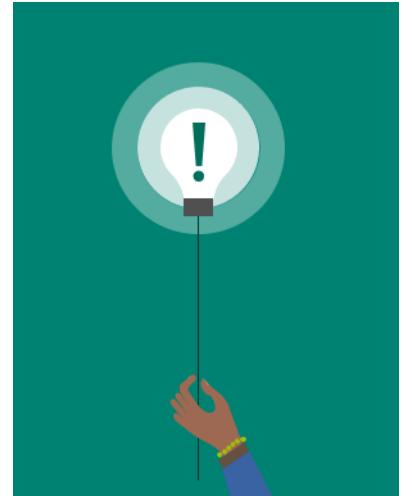
What is the primary tool used for Power BI Report development?

# Lesson 2: Connection to Data

# Objectives

After completing this learning, you will be able to:

- Describe different data sources for Power BI
- Describe different data connectivity modes



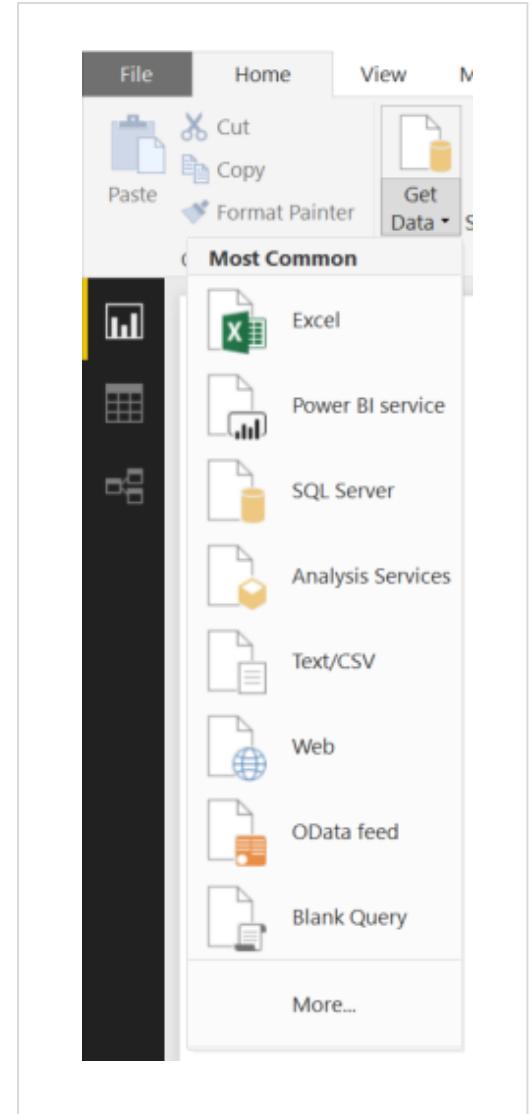
# Connecting Data - Prepare

The “**Get Data**” option allows the definition of **connections** to data sources and the **selection of entities** and columns

**Authentication** options can be specified including **credential types** such as Windows or others

Data can be **imported in-memory** and/or be **accessed live** for some data sources

Each selected entity will become a query that can be further enhanced to meet the business requirements



# Connecting Data - Prepare

## Supported data sources (sample list)

File	Database	Azure	Other/Online Services	Other	
<ul style="list-style-type: none"><li>• Excel</li><li>• Text/CSV</li><li>• XML</li><li>• JSON</li><li>• Folder</li><li>• SharePoint Folder</li><li>• PDF</li></ul>	<ul style="list-style-type: none"><li>• SQL Server (<b>DQ</b>)</li><li>• Access</li><li>• SQL Server Analysis Services (<b>Live Connection</b>)</li><li>• Oracle (<b>DQ</b>)</li><li>• IBM DB2 (<b>DQ</b>)</li><li>• IBM Informix (Beta)</li><li>• IBM Netezza (<b>DQ</b>)</li><li>• MySQL</li><li>• PostgreSQL(<b>DQ</b>)</li><li>• Sybase</li><li>• Teradata (<b>DQ</b>)</li><li>• SAP HANA (<b>DQ</b>)</li><li>• SAP BW V2 (<b>DQ</b>)</li><li>• SAP BW Message Server connector</li><li>• Amazon Redshift (<b>DQ</b>)</li><li>• Impala (Beta) (<b>DQ</b>)</li><li>• Google BigQuery (Beta) (<b>DQ</b>)</li><li>• MarkLogic</li><li>• Snowflake (<b>DQ</b>)</li></ul>	<ul style="list-style-type: none"><li>• SQL Database (<b>DQ</b>)</li><li>• SQL Data Warehouse (<b>DQ</b>)</li><li>• Analysis Services (Beta) (<b>Live Connection</b>)</li><li>• Blob Storage</li><li>• Table Storage</li><li>• Cosmos DB (Beta)</li><li>• Data Lake Store</li><li>• HDInsight (HDFS)</li><li>• HDInsight Spark (<b>DQ</b>)</li></ul>	<ul style="list-style-type: none"><li>• Power BI Service</li><li>• SharePoint Online List</li><li>• Exchange Online</li><li>• Dynamics 365 (online)</li><li>• Dynamics 365 for Financials (Beta)</li><li>• Dynamics NAV</li><li>• Dynamics 365 Business Central (On Premise) Connectors</li><li>• Common Data Service (Beta)</li><li>• Azure Consumption Insights (Beta)</li><li>• Visual Studio Team Services (Beta)</li><li>• Salesforce Objects</li><li>• Salesforce Reports</li><li>• Google Analytics</li><li>• appFigures (Beta)</li><li>• Dynamics 365 for Customer Insights (Beta)</li><li>• Product Insights</li><li>• Asana</li><li>• TIBCO</li></ul>	<ul style="list-style-type: none"><li>• Facebook</li><li>• GitHub (Beta)</li><li>• Kusto (Beta)</li><li>• MailChimp (Beta)</li><li>• Marketo (Beta)</li><li>• Mixpanel (Beta)</li><li>• Planview Enterprise (Beta)</li><li>• Projectplace (Beta)</li><li>• QuickBooks Online (Beta)</li><li>• Smartsheet</li><li>• SparkPost (Beta)</li><li>• SQL Sentry (Beta)</li><li>• Stripe (Beta)</li><li>• SweetIQ (Beta)</li><li>• Troux (Beta)</li><li>• Twilio (Beta)</li><li>• tyGraph (Beta)</li><li>• Webtrends (Beta)</li><li>• ZenDesk (Beta)</li><li>• Data.World (Beta)</li><li>• Adobe Analytics</li></ul>	<ul style="list-style-type: none"><li>• Vertica</li><li>• Web</li><li>• SharePoint List</li><li>• OData Feed</li><li>• Active Directory</li><li>• Microsoft Exchange</li><li>• Hadoop File (HDFS)</li><li>• Spark (<b>DQ</b>)</li><li>• R Script</li><li>• ODBC</li><li>• OLE DB</li><li>• Blank Query</li><li>• LinkedIn Sales Navigator (beta)</li><li>• Vena</li><li>• MicroStrategy</li><li>• FHIR</li><li>• FactSet</li><li>• Jamf Pro</li><li>• Cognite</li></ul>

The available data sources are *constantly evolving* and some of them support DirectQuery (DQ). No data is imported to Power BI in DQ and Live Connections. Can seamlessly combine data from one or more DQ sources, and/or combine data from a mix of DQ sources and import data through Composite Models. There are some limitations on the transformations that can be done with DQ.

# Connecting Data - Prepare

Three ways to connect to your data

	Import	DirectQuery	Live/Exploration
Overview	<ul style="list-style-type: none"><li>• ETL</li><li>• Data download</li></ul>	<ul style="list-style-type: none"><li>• Select specific tables</li><li>• No data download</li><li>• Queries triggered from Report visuals</li></ul>	<ul style="list-style-type: none"><li>• Explore source objects from Report surface</li><li>• No data download</li><li>• Queries triggered from Report visuals</li></ul>
Supported Data Sources	<ul style="list-style-type: none"><li>• All sources (&gt;80 sources)</li></ul>	<ul style="list-style-type: none"><li>• SQL Server</li><li>• Azure SQL Database</li><li>• Azure SQL Data Warehouse</li><li>• SAP HANA</li><li>• Oracle</li><li>• Teradata</li></ul>	<ul style="list-style-type: none"><li>• SQL Server Analysis Services (Tabular &amp; Multidimensional)</li><li>• Azure Analysis Services</li></ul>
Max # of data sources per report	<ul style="list-style-type: none"><li>• Unlimited</li></ul>	<ul style="list-style-type: none"><li>• Multiple</li></ul>	One
Data Transformations	<ul style="list-style-type: none"><li>• All transformations (100's)</li></ul>	<ul style="list-style-type: none"><li>• Partial support (varies by data source)</li></ul>	None
Mashup Capabilities	<ul style="list-style-type: none"><li>• Merge (Joins)</li><li>• Append (Union)</li><li>• Parameterized queries</li></ul>	<ul style="list-style-type: none"><li>• Merge (Joins)</li><li>• Append (Union)</li></ul>	None
Modeling Capabilities	<ul style="list-style-type: none"><li>• Relationships</li><li>• Calculated Columns &amp; Tables</li><li>• Measures</li><li>• Hierarchies</li><li>• Time Intelligence</li></ul>	<ul style="list-style-type: none"><li>• Relationships</li><li>• Calculated Columns</li><li>• Measures</li><li>• Hierarchies</li><li>• Limited Time Intelligence</li></ul>	<ul style="list-style-type: none"><li>• Measures</li></ul>

# Connecting Data - Prepare

## Direct Query - Overview

### What is Direct Query?

- **Connectivity Method** in Power BI and SSAS where the modeler defines the model but doesn't import any data
- **Any data** needed for visualizations will be retrieved **from the data source directly**

### Major Benefits of Direct Query

- **No need to move any data**, data stays at the source where it is created offering real time updates whenever changes are made.
- Ability to leverage the **security** as it is defined in the data source when using Single Sign On (SSO).
- **You will not hit any memory limits in Power BI** or SSAS so you can see all the raw data (all transactions)

# Connecting Data - Prepare

## Direct Query - Pitfalls

### OLTP Data Source – not a Data Warehouse

- Creating a decent model - more manageability and query complexity (Which could lead to performance issues).
- Analytical query patterns (e.g. dimensional joins) might be performance issues

### MPP Data Source - Big Data solution

- High concurrency might be an issue (the reports are consumed by hundreds of users at the peak time)

### Security

- The security as set up in the transactional data source might not be the same security as you expect for reporting

# Connecting Data - Prepare

Direct Query – Best practice

## Required Investments to use Direct Query

- **Optimizing your data source** with best practice or tuning techniques like In-memory indexes designed for query to load data into reports.
- Instead of views, you **create new analytical tables** to shape the data and **add specific performance features**.
- **Leverage a fast data solution** like HANA, SQL DW, Netezza, Spark, etc. In each case **test the load to determine the amount of concurrency**.

## Tradeoffs – DQ or Import?

- Data refresh - SSAS partitioning
- Memory limit - Smart compression
- Security

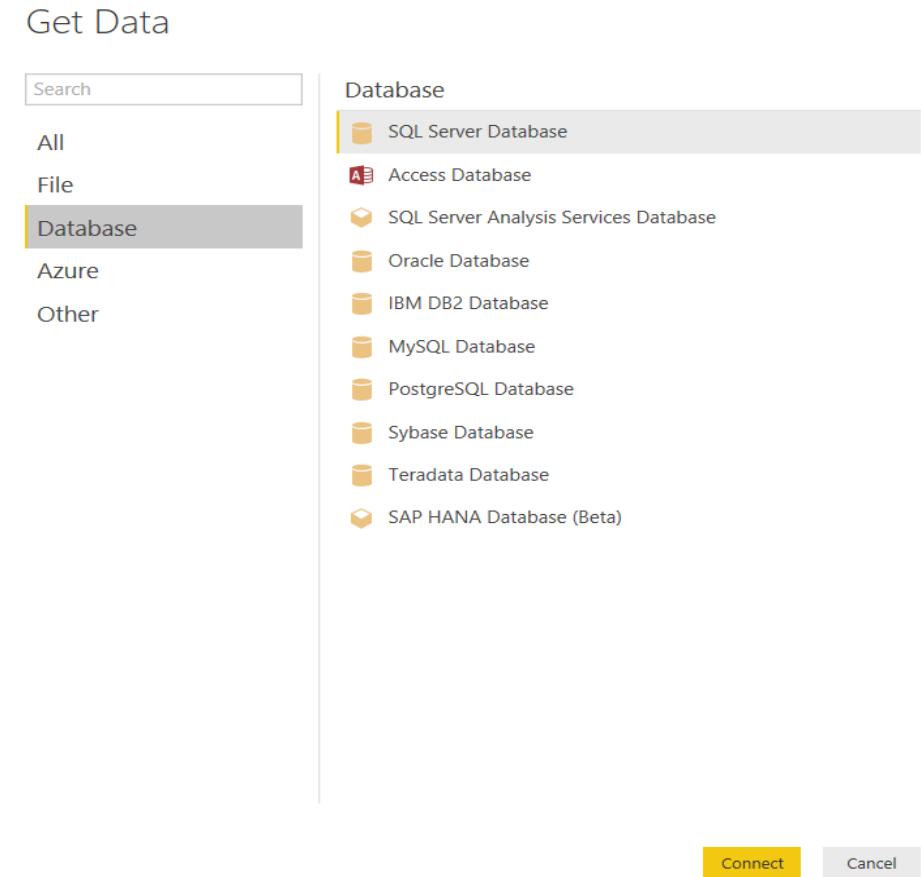
# Connecting Data - Prepare

Connecting to SQL Server (an example)

Access the “**Get Data**” button  
from the top ribbon

Select **SQL Server Database** from the  
“**Database**” type of sources

And then click “**Connect**”



# Connecting Data - Prepare

Connecting to SQL Server (an example)

Specify an instance to connect to

Optionally, specify a **database** – if not, a list will be presented

Specify **Data Connectivity** mode

Advanced options includes **Timeout, query, failover support option**, etc.

SQL Server database

Server i  
[REDACTED]\SQL2017

Database (optional)  
[REDACTED]

Data Connectivity mode i

- Import  
 DirectQuery

► Advanced options

**OK** Cancel

# Connecting Data - Prepare

Connecting to SQL Server (an example)

A **DirectQuery** enabled data source will include both “Import” and “DirectQuery” option in connection setting dialog

DirectQuery allows querying **very large datasets** and with a **live connection** (no data is imported)

**Import brings all the data into the Desktop**

Data Connectivity mode 

Import

DirectQuery

 Advanced options

# Connecting Data - Prepare

Connecting to SQL Server (an example)

---

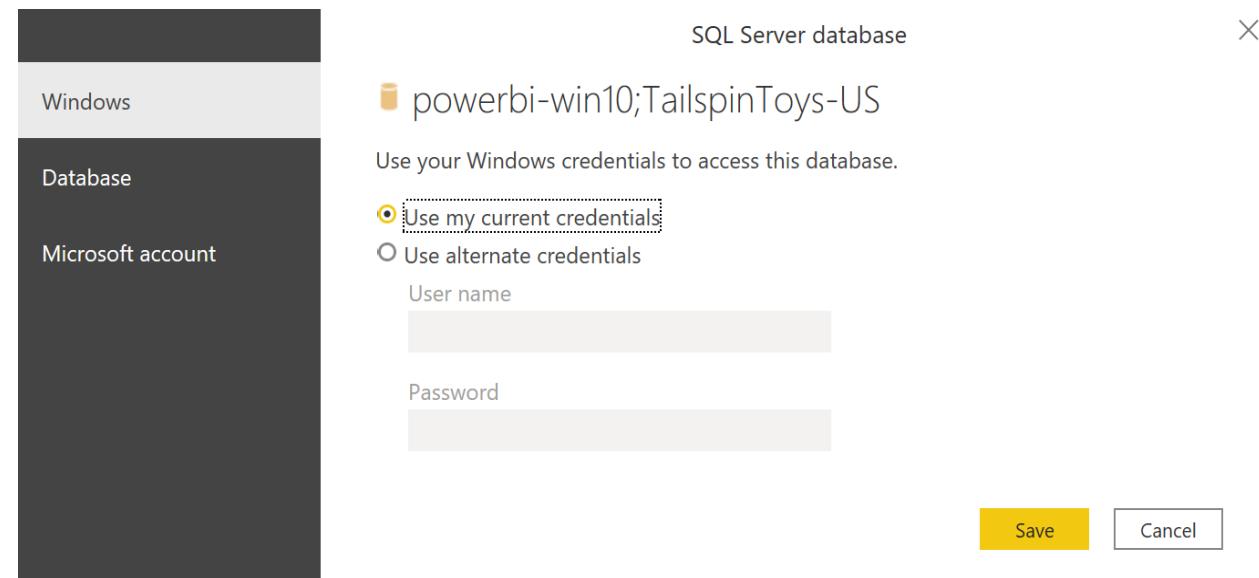
Select the right authentication type for your data source

---

Authentication options will depend on the connector used

---

Use Windows or Database credentials for on premises SQL databases



# Connecting Data - Prepare

Connecting to SQL Server (an example)

---

Now **select** the desired tables

---

A **preview** is generated for ease of use

---

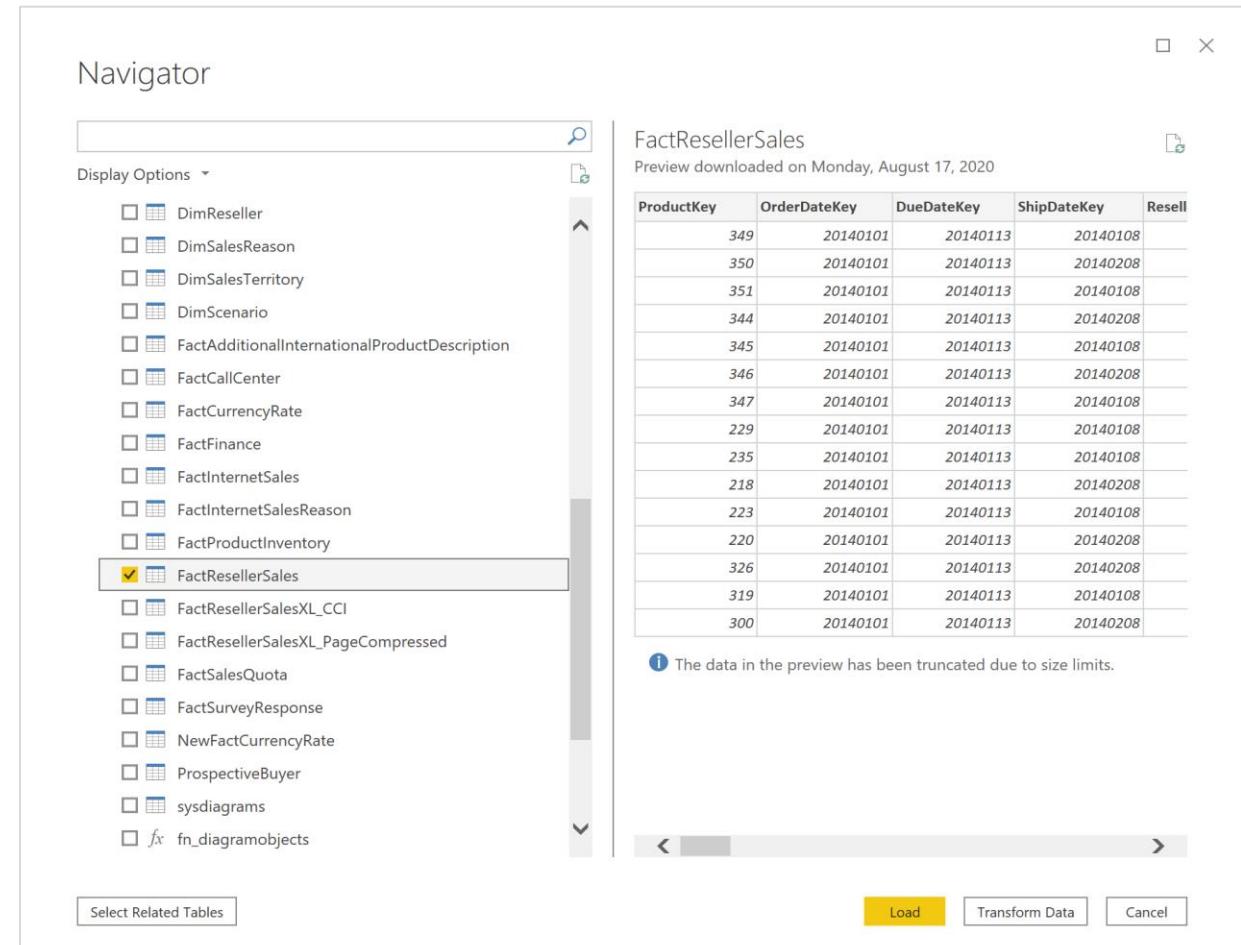
You can use **relationship detection** per-table

---

Choose “**Load**” if the data is ready

---

Choose “**Transform Data**” if the data needs to be transformed



The screenshot shows the Power BI Data Flow 'Navigator' interface. On the left, a list of tables is displayed with 'FactResellerSales' selected. The main area shows a preview of the 'FactResellerSales' table data, which includes columns: ProductKey, OrderDateKey, DueDateKey, ShipDateKey, and Resell. The data preview shows approximately 20 rows of sales records. A note at the bottom states: "The data in the preview has been truncated due to size limits." At the bottom right, there are buttons for 'Load', 'Transform Data', and 'Cancel'.

ProductKey	OrderDateKey	DueDateKey	ShipDateKey	Resell
349	20140101	20140113	20140108	
350	20140101	20140113	20140208	
351	20140101	20140113	20140108	
344	20140101	20140113	20140208	
345	20140101	20140113	20140108	
346	20140101	20140113	20140208	
347	20140101	20140113	20140108	
229	20140101	20140113	20140108	
235	20140101	20140113	20140108	
218	20140101	20140113	20140208	
223	20140101	20140113	20140108	
220	20140101	20140113	20140208	
326	20140101	20140113	20140208	
319	20140101	20140113	20140108	
300	20140101	20140113	20140208	

# Connecting Data - Prepare

Connecting to Microsoft SQL Server (an example)

Use Query Editor to further refine our data, if we specified “Transform Data”:

The screenshot shows the Power BI Query Editor interface. On the left, there's a sidebar titled "8 Queries" containing a list of tables: FactInternetSales, DimCurrency, DimCustomer, DimDate, DimProduct, DimPromotion, DimSalesTerritory, and FactInternetSalesReason. The main area displays a table titled "AdventureworksDW2014[[Schema="dbo", Item="FactInternetSales"]][Data]" with columns: ProductKey, OrderDateKey, DueDateKey, ShipDateKey, CustomerKey, PromotionKey, CurrencyKey, SalesTerritoryKey, SalesOrderNumber, and SalesOrderLineNumber. The table contains approximately 999+ rows of data. At the top, the ribbon has tabs like File, Home, Transform, Add Column, and View. The Transform tab is selected, showing various tools: Close & Apply, New Query, Refresh Preview, Advanced Editor, Properties, Choose Columns, Remove Columns, Keep Rows, Remove Rows, Remove Duplicates, Remove Errors, Sort, Split Column, Group By, Replace Values, Merge Queries, Append Queries, Combine Binaries, and Combine. To the right of the table, a "Query Settings" pane is open, showing "PROPERTIES" (Name: FactInternetSales) and "APPLIED STEPS" (Source: Navigation). The bottom status bar indicates "35 COLUMNS, 999+ ROWS" and "PREVIEW DOWNLOADED AT 11:51".

# Connecting Data - Prepare

Connecting to SQL Server (an example)

---

Finally, we would apply our changes so that we can start exploring the data

---

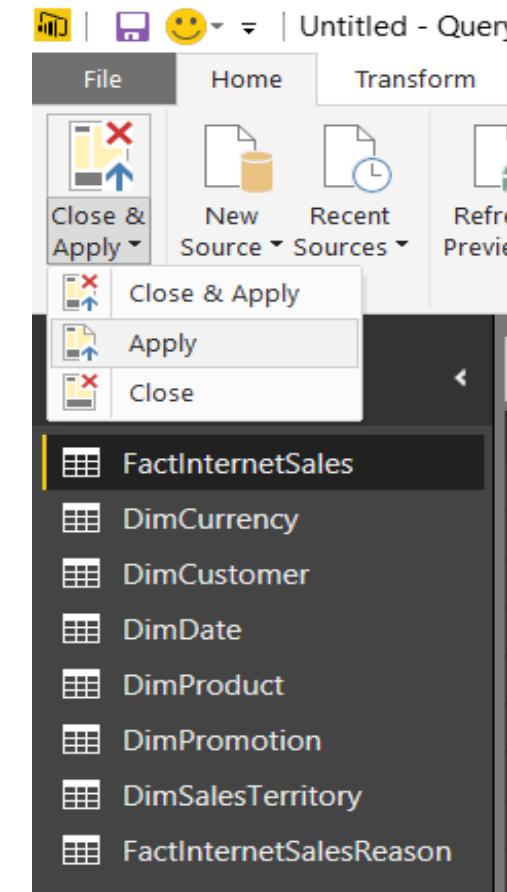
**Close & Apply** – Closes the query editor and loads the data

---

**Apply** – Re-loads the data from source

---

**Close** – Closes the query editor without applying changes



# Connecting Data - Prepare

Changing connections

---

After the **connections** are established,  
you can easily change them

---

For instance, you can change from Test  
to Production Environment

---

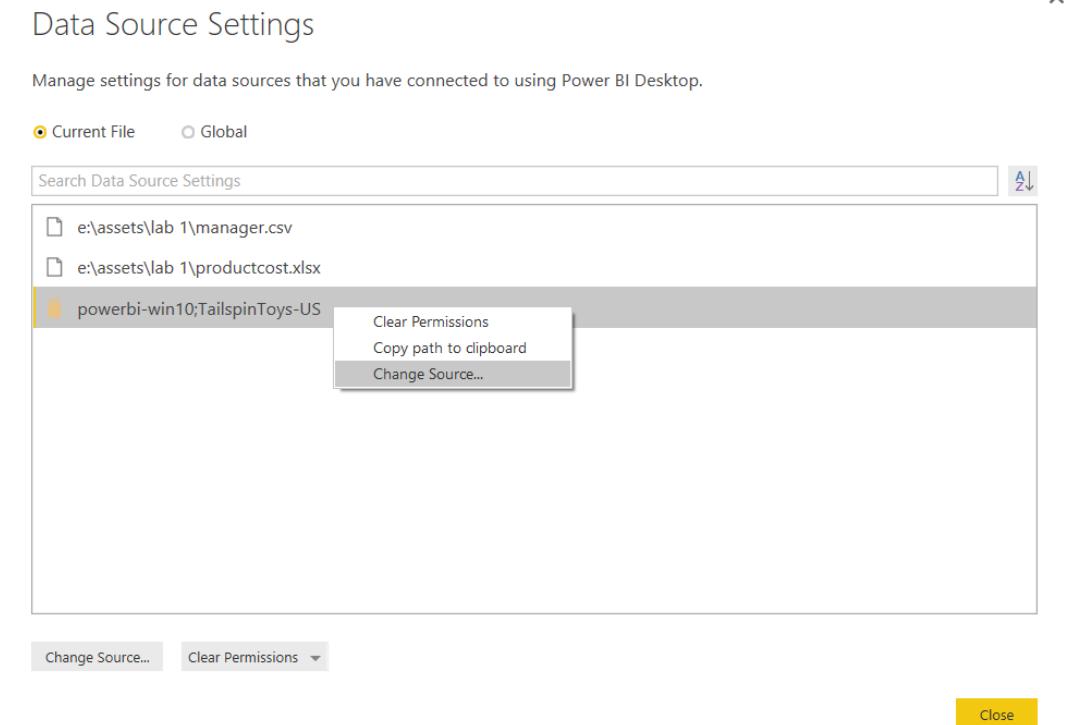
**Go to File -> Options and Settings -  
> Data Source Settings**

---

Specified credentials can be cleared

---

The scope can be the current file or  
previous created connections



# Connecting Data - Prepare

.pbids files

Streamlines "Get Data".

Only one source is allowed

JSON Format  
(requires escape characters)

## SQL Server

```
JSON

{
  "version": "0.1",
  "connections": [
    {
      "details": {
        "protocol": "tds",
        "address": {
          "server": "server-name-here",
          "database": "db-name-here (optional)"
        },
        "options": {},
        "mode": "DirectQuery"
      }
    ]
}
```

# Connecting Data - Prepare

Power BI Custom Connectors

---

**Create** your own **data connector** which can be added to the Get Data menu.

---

Connect to a **data source that may not be supported yet** by Power BI Desktop or **simplify access** to data marts or other sources

---

Use your **own logic**, and use **multiple options for credentials**, such as Windows, basic, API Key, and database authentication; and more.

---

The connector can work in both **Import** or **DirectQuery** and works with the **gateway**.

---

Created using the same **M language** used by Power Query; (Power Query SDK is available for Visual Studio)

# Connecting Data - Prepare

## Power BI Custom Connectors

Some of these connectors will be **certified by Microsoft**

A **security parameter** dictates if non-certified custom connectors can be used

Be **mindful** of the extensions you run

The screenshot shows the 'Get Data' interface in Power BI. On the left, there's a sidebar with categories like All, File, Database, Power BI, Azure, Online Services, and Other. Below these are sections for OData Feed, Active Directory, Microsoft Exchange, Hadoop File (HDFS), Spark, R script, ODBC, and OLE DB. A large list of connectors follows, including Denodo, Dremio (Beta), Exasol (Beta), Jethro (Beta), Kyligence Enterprise (Beta), Paxata (Beta), and TeamDesk (Beta). At the bottom of this list is a red box labeled 'Certified Connectors'. On the right, a modal window titled 'Options' is open. It has tabs for GLOBAL, Security (which is selected and highlighted in yellow), Privacy, Updates, Usage Data, and Diagnostics. Under the GLOBAL tab, there are sections for Native Database Queries (with a checked checkbox for 'Require user approval for new native database queries') and Web Preview Warning Level (set to 'Moderate'). The Security tab contains sections for Data Extensions, where a radio button is selected for '(Recommended) Only allow certified extensions to load'. A red box highlights this section. At the bottom of the Security tab, there's a link 'Learn more about data extensions'.

**Questions?**



# Knowledge Check

What is the primary difference between the two data connectivity modes in Power BI - Direct Query (DQ) and Import?

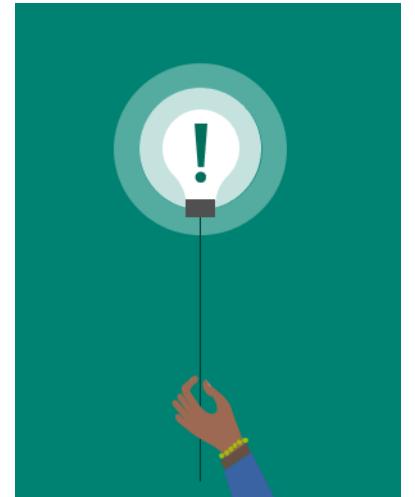
Name some connectors that support Direct Query connection

# Lesson 3: Shaping Data

# Objectives

After completing this learning, you will be able to:

- Describe shaping data to meet business requirements
- Learn different data transformation methods in Power BI



# Shaping Data

---

Shaping is the act of **transforming** the data to meet our **business requirements**

---

Apply **data cleaning operations** and correct/remove errors

---

**Rename the data** so it is more meaningful

---

**Combine data** from different data sources in a single table

---

Create **personal** (reusable) **views** of the data

---

A set of **out-of-the box transformations** are available to help

---

A **programming language** is available for addressing more complex use-cases

---

**Source data is not changed**, all the transformations are applied on the Desktop engine

# Shaping Data

## Power Query Editor

After the data is loaded or referenced into Power BI Desktop, the “**Transform Data**” option in the “**Home**” ribbon allows the edition of the imported entities

**Each entity** that was loaded from each data source will generate a **separate query**

**Transformations** are applied **per-entity**, although it is possible to combine them.

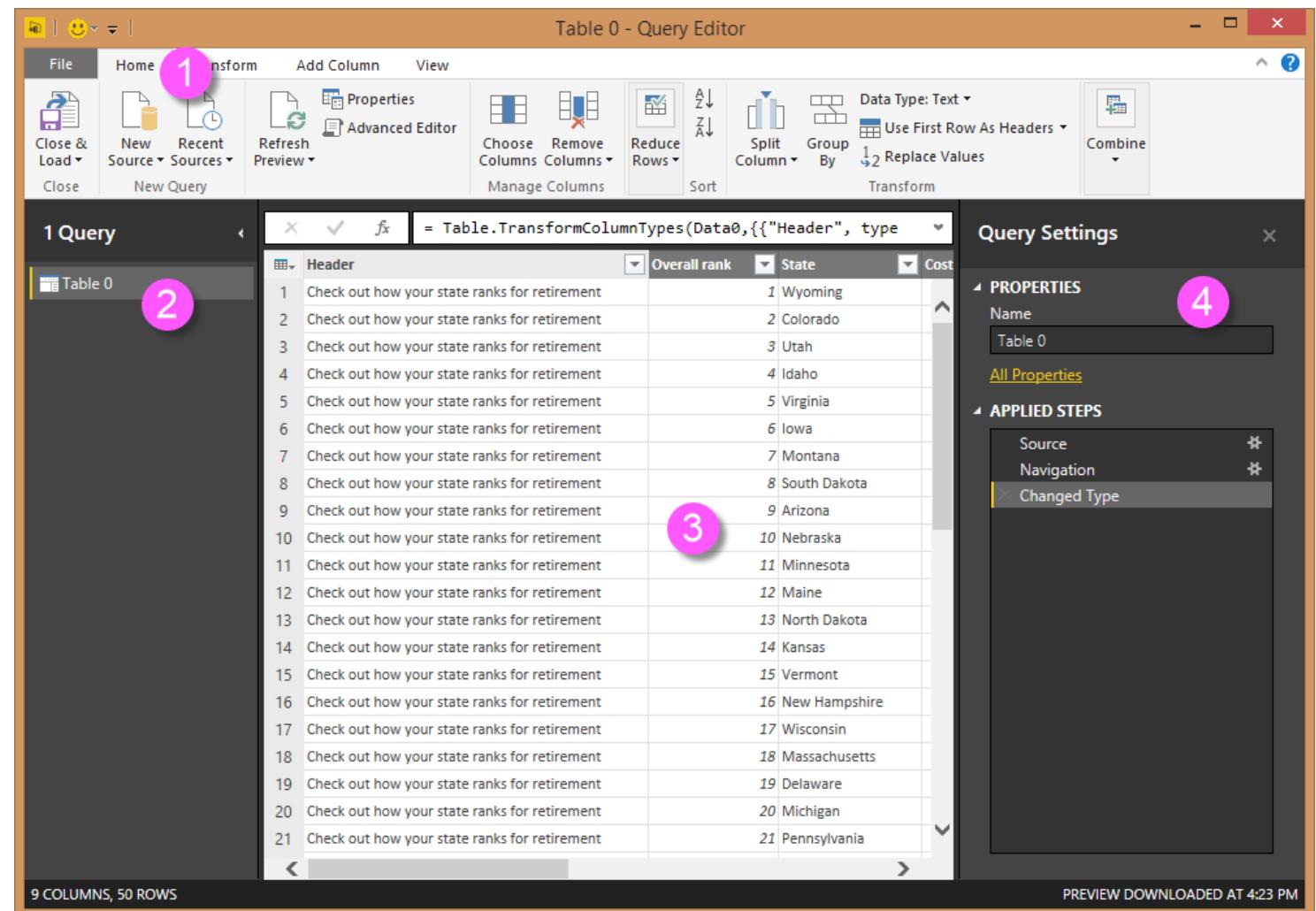
The screenshot shows the Power BI Desktop interface. The ribbon is visible at the top, with the "Home" tab selected. On the far right of the ribbon, there is a "Transform data" button, which is highlighted with an orange box. The main workspace on the right displays a list titled "8 Queries" containing items like "FactInternetSales", "DimCurrency", etc. Below the list is a preview table with two columns: "ProductKey" and "OrderID". The first few rows of the table are visible, showing values such as 1, 2, 3, 4, 5, 6, 7, 8, and 9.

ProductKey	OrderID
1	310
2	346
3	346
4	336
5	346
6	311
7	310
8	351
9	344

# Shaping Data

## Power Query Editor

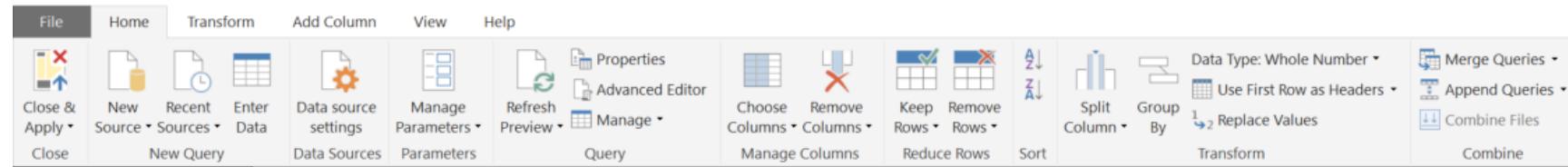
- 1 Query Ribbon
- 2 The Left Pane
- 3 The Center Pane
- 4 The Query Settings Pane



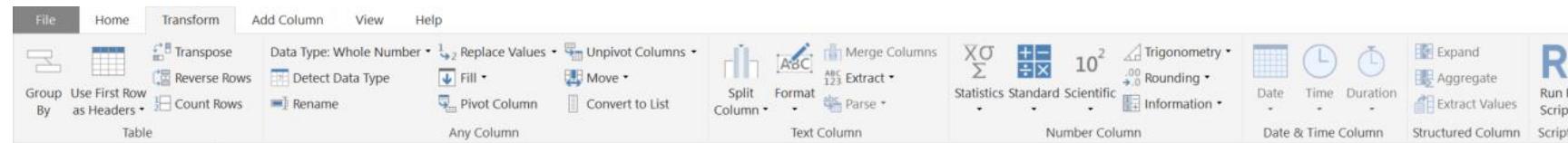
# Shaping Data

## Power Query Editor – Ribbon

- The “**Home**” tab contains the common query tasks including the combination of queries



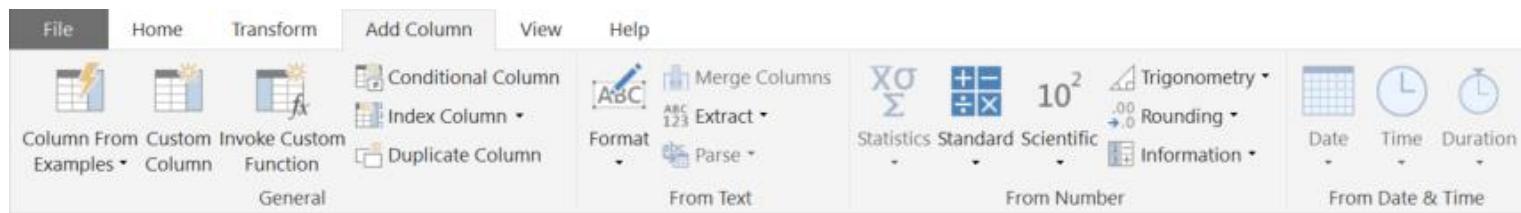
- The “**Transform**” tab provides access to **common data transformation tasks**, such as adding or removing columns, changing data types, splitting columns, and other data-driven tasks



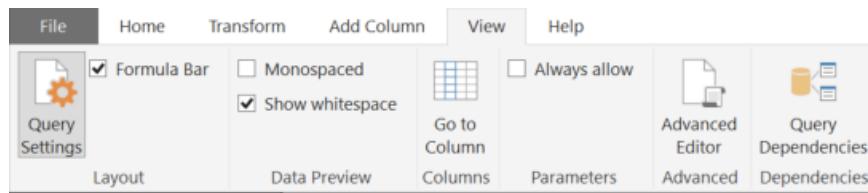
# Shaping Data

## Power Query Editor – Ribbon

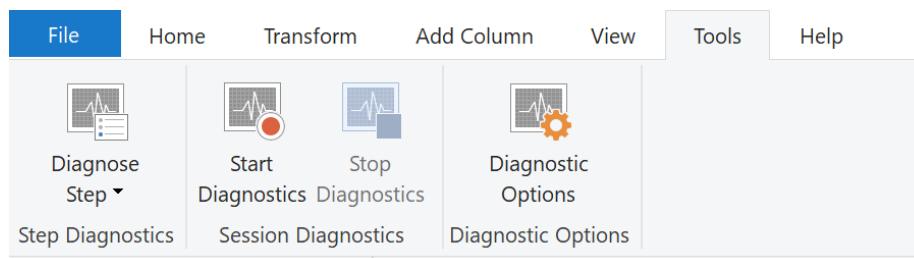
- The “**Add Column**” tab provides additional tasks associated with columns such as formatting column data, adding custom columns or invoking functions



- The “**View**” tab provides access to **query settings** and the **Advanced Editor** where we can develop our own data transformation scripts



- The “**Tools**” tab provides tools for query diagnostics



# Shaping Data

Power Query Editor – The Left Pane

---

Displays the **number of active queries**

---

The **name** of each query

---

Clicking on a query displays the query preview on the Center Pane

---

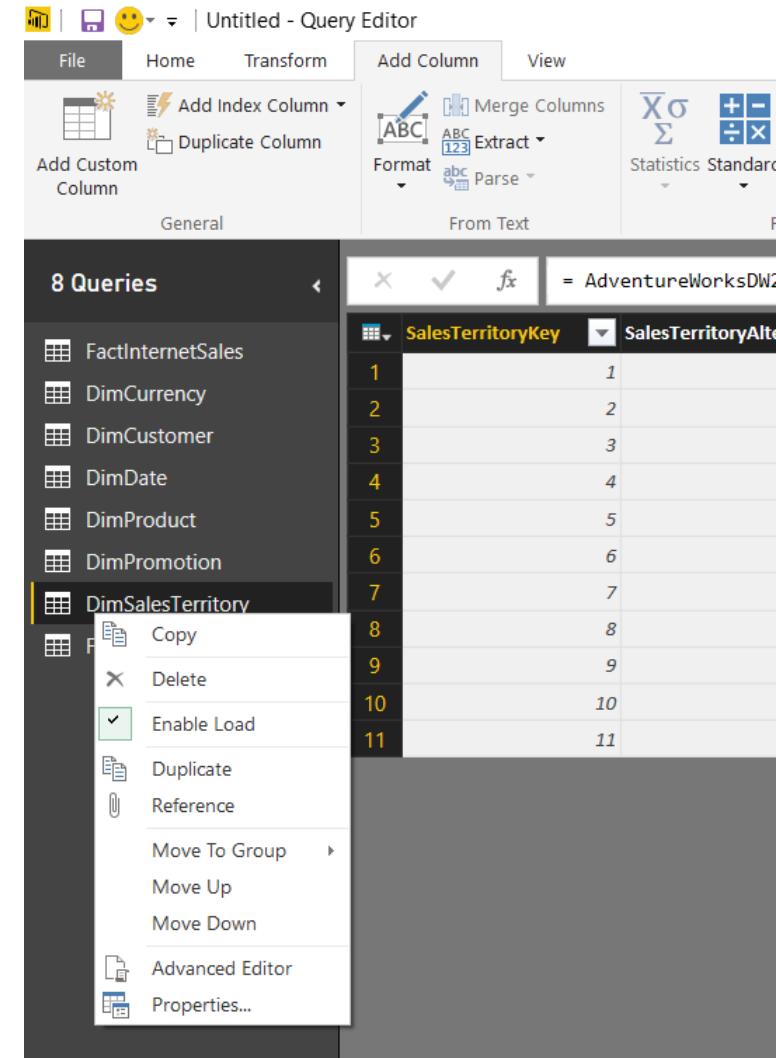
Also allows to **disable the query load**

---

**Organizing** the queries into groups

---

Options for **duplicating** or creating a **reference**



# Shaping Data

Power Query Editor – The Center Pane

Displays the **data for the selected query and the current transformation step**

Many of the transformations can be directly applied here, namely, column related

It displays a **preview of the data** and might not display the entire dataset

The screenshot shows the Power Query Editor's Center Pane. On the left, the '8 Queries' pane lists several queries, with 'FactInternetSales' selected. The main area displays a preview of the 'FactInternetSales' data, showing columns like ProductKey, OrderDateKey, DueDateKey, ShipDateKey, Custom, CurrencyKey, SalesTerritoryKey, SalesOrderNumber, and SalesOrderLineNumber. A context menu is open over the 'Custom' column, with 'Remove Other Columns' highlighted. To the right, the 'Query Settings' pane shows 'Properties' (Name: FactInternetSales) and 'Applied Steps' (Source, Navigation).

ProductKey	OrderDateKey	DueDateKey	ShipDateKey	Custom	CurrencyKey	SalesTerritoryKey	SalesOrderNumber	SalesOrderLineNumber
1	310	20101229	20110110	20110105	19	6	SO43697	
2	346	20101229	20110110	20110105	39	7	SO43698	
3	346	20101229	20110110	20110105	100	1	SO43699	
4	336	20101229	20110110	20110105	100	4	SO43700	
5	346	20101229	20110110	20110105	6	9	SO43701	
6	311	20101230	20110111	20110106	100	4	SO43702	
7	310	20101230	20110111	20110106	6	9	SO43703	
8	351	20101230	20110111	20110106	6	9	SO43704	
9	344	20101230	20110111	20110106	6	9	SO43705	
10	312	20101231	20110112	20110107	100	4	SO43706	
11	312	20101231	20110112	20110107	100	4	SO43707	
12	330	20101231	20110112	20110107	98	10	SO43708	
13	313	20101231	20110112	20110107	6	9	SO43709	
14	314	20101231	20110112	20110107	6	9	SO43710	
15	314	20110101	20110113	20110108	100	1	SO43711	
16	311	20110101	20110113	20110108	29	8	SO43712	
17	310	20110102	20110114	20110109	100	4	SO43713	
18	311	20110102	20110114	20110109	98	10	SO43714	
19	314	20110102	20110114	20110109	6	9	SO43715	
20	311	20110102	20110114	20110109	16529	1	SO43716	
21	336	20110102	20110114	20110109	25249	1	SO43717	
22	311	20110103	20110115	20110110	27668	1	100	SO43718

# Shaping Data

## Power Query Editor – The Query Settings Pane

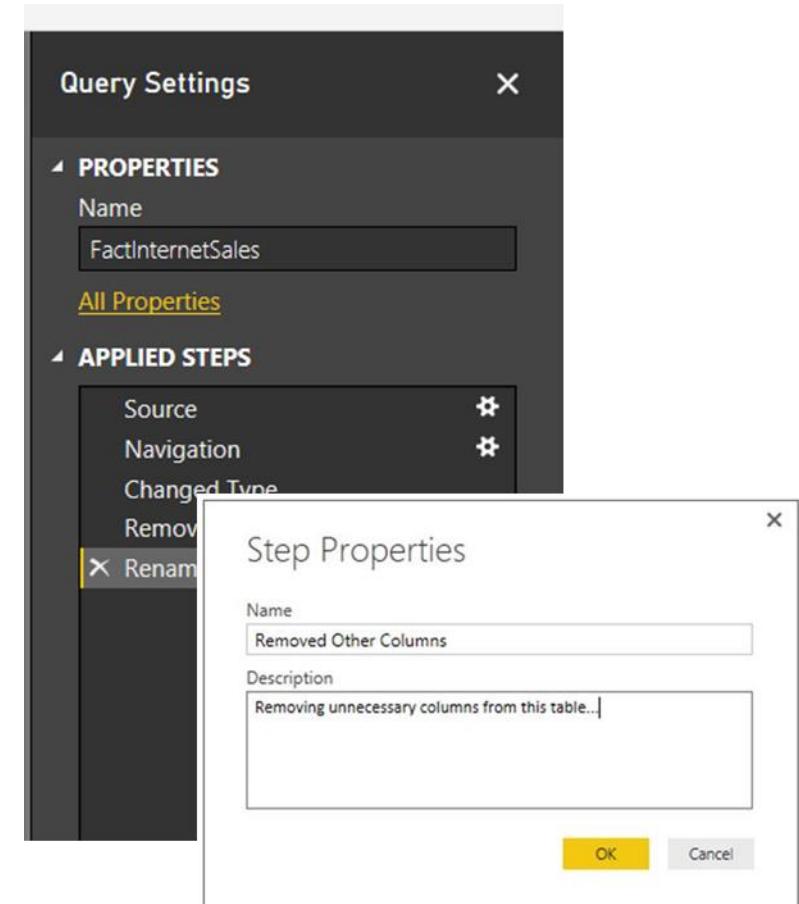
Each transformation that is applied to a query generates a **step**

Those **steps are applied in order** to generate the desired result

Order can be changed if it respects the needs of previous steps

It is also possible to **edit each step's properties** via this pane

Possible to **add Description** to Query steps



# Demonstration

## Building a Query

- Query Features Inside of Power BI Desktop



# Shaping Data

Available Data Transformations

## Transform

- Manage columns
- Reduce rows
- Merge/Split columns
- Group rows in a table
- Aggregate data from a table
- Use first Row as headers
- Replace Values
- Transpose
- Reverse Rows
- Count Rows
- Conditional Columns
- Filter Table
- Pivot/Unpivot
- Change Data Types
- Text/Number/Date Formatting
- Extract parts of strings
- Fill Rows
- Expand Rows from a related table
- Merge Queries
- Append Queries
- Combine Binaries

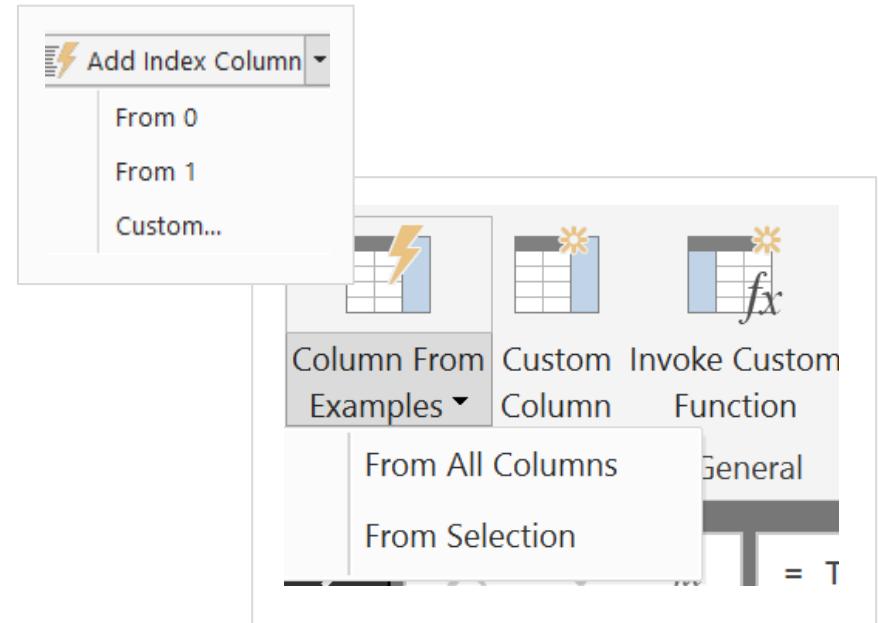
**And more ...**

# Shaping Data

## Available Data Transformations

### Add Column

- Insert Custom Column (also used to invoke functions)
- Insert Index Column
- Duplicate Column
- Column From Examples



# Shaping Data

## Available Data Transformations

### Column From Examples

- Useful when the result is known, but not the transformations to apply or where they are
- Generates an expression automatically (like Excel's Flash Fill)

The screenshot shows the 'Add Column From Examples' dialog box in Power BI. At the top, there's a toolbar with various data transformation icons like 'Conditional Column', 'Merge Columns', and 'Format'. Below the toolbar is a section titled 'Add Column From Examples' with instructions: 'Enter sample values to create a new column (Ctrl+Enter to apply)'. A 'Transform' formula is displayed: `Text.Combine({{"State & postal abbreviation"}, " was founded in ", Text.From([Year from ParseDate]), " and has ", "#Replaced Population (2016 est)[18]"}, " people")`. The main area is a table titled 'Queries' with columns: 'Population (2016 est)[18]', 'Corrected State', 'Replaced Population (2016 est)[18]', 'ParseDate', and 'Combined'. The table lists 18 rows of US state data. To the right of the table, the 'Combined' column shows the resulting text output for each state, such as 'Alabama, was founded in 1819 and has 48'.

Population (2016 est)[18]	Corrected State	Replaced Population (2016 est)[18]	ParseDate	Combined
4,863,300	Alabama	4863300	14/12/1819	Alabama, was founded in 1819 and has 48
741,894	Alaska	741894	03/01/1959	Alaska, was founded in 1959 and has 741894
6,931,071	Arizona	6931071	14/02/1912	Arizona, was founded in 1912 and has 6931071
2,988,248	Arkansas	2988248	15/06/1836	Arkansas, was founded in 1836 and has 2988248
39,250,017	California	39250017	09/09/1850	California, was founded in 1850 and has 39250017
5,540,545	Colorado	5540545	01/08/1876	Colorado, was founded in 1876 and has 5540545
3,576,452	Connecticut	3576452	09/01/1788	Connecticut, was founded in 1788 and has 3576452
952,065	Delaware	952065	07/12/1787	Delaware, was founded in 1787 and has 952065
20,612,439	Florida	20612439	03/03/1845	Florida, was founded in 1845 and has 20612439
10,310,371	Georgia	10310371	02/01/1788	Georgia, was founded in 1788 and has 10310371
1,428,557	Hawaii	1428557	21/08/1959	Hawaii, was founded in 1959 and has 1428557
1,683,140	Idaho	1683140	03/07/1890	Idaho, was founded in 1890 and has 1683140
12,801,539	Illinois	12801539	03/12/1818	Illinois, was founded in 1818 and has 12801539
6,633,053	Indiana	6633053	11/12/1816	Indiana, was founded in 1816 and has 6633053
3,134,693	Iowa	3134693	28/12/1846	Iowa, was founded in 1846 and has 3134693
2,907,289	Kansas	2907289	29/01/1861	Kansas, was founded in 1861 and has 2907289
4,436,974	Kentucky	4436974	01/06/1792	Kentucky, was founded in 1792 and has 4436974
...	...	...	...	...

# Shaping Data

## Available Data Transformations – Machine Learning Scripts

Run R or Python Scripts from the Query Editor

Perform **data cleansing, advanced data shaping, and analytics** including completion of missing data, predictions, and clustering etc.

R or Python engine should be **installed locally**

The screenshot shows the Power Query Editor interface with the 'Transform' tab selected. On the ribbon, there is a red box around the 'Run R' and 'Run Python' buttons. A modal dialog box titled 'Run R Script' is open, containing an R script for data completion:

```
# 'dataset' holds the input data for this script
library(mice)
tempData <- mice(dataset, m=1, maxit=50, meth='pmm', seed=100)
completedData <- complete(tempData, 1)
output <- dataset
output$CompletedValues <- completedData$"SMI missing values"
```

The dialog also displays a message about the R home directory and configuration options.

# Shaping Data

Available Data Transformations – Advanced Editor

Every transformation generates code automatically

For more advanced scenarios

OOB transformations will cover more than 90% of the cases



The screenshot shows the Power BI Advanced Editor window titled "FactInternetSales". The editor displays a block of DAX code:

```
let
    Source = Sql.Databases("testbot01.cloudapp.net,60000\SQL2K14"),
    AdventureWorksDW2014 = Source{[Name="AdventureWorksDW2014"]}[Data],
    dbo_FactInternetSales = AdventureWorksDW2014{[Schema="dbo",Item="FactInternetSales"]}[Data],
    #"Changed Type" = Table.TransformColumnTypes(dbo_FactInternetSales,{{"OrderQuantity", Int64.Type}}),
    #"Removed Columns" = Table.RemoveColumns(#"Changed Type",{"DimCurrency", "DimCustomer", "DimDate(DueDateKey)", "DimDate(OrderDateKey)", "DimDate(ShipDateKey)", "DimProduct", "DimPromotion"}),
    #"Renamed Columns" = Table.RenameColumns(#"Removed Columns",{{"SalesOrderNumber", "Sales Order Number"}}),
    #"Merged Queries" = Table.NestedJoin(#"Renamed Columns", {"ProductKey"}, DimProduct, {"ProductKey"}, "NewColumn", JoinKind.LeftOuter),
    #"Expanded NewColumn" = Table.ExpandTableColumn(#"Merged Queries", "NewColumn", {"ListPrice"}, {"NewColumn.ListPrice"})
in
    #"Expanded NewColumn"
```

At the bottom of the code editor, a message states: "No syntax errors have been detected." The window includes standard controls for minimizing, maximizing, and closing the application.

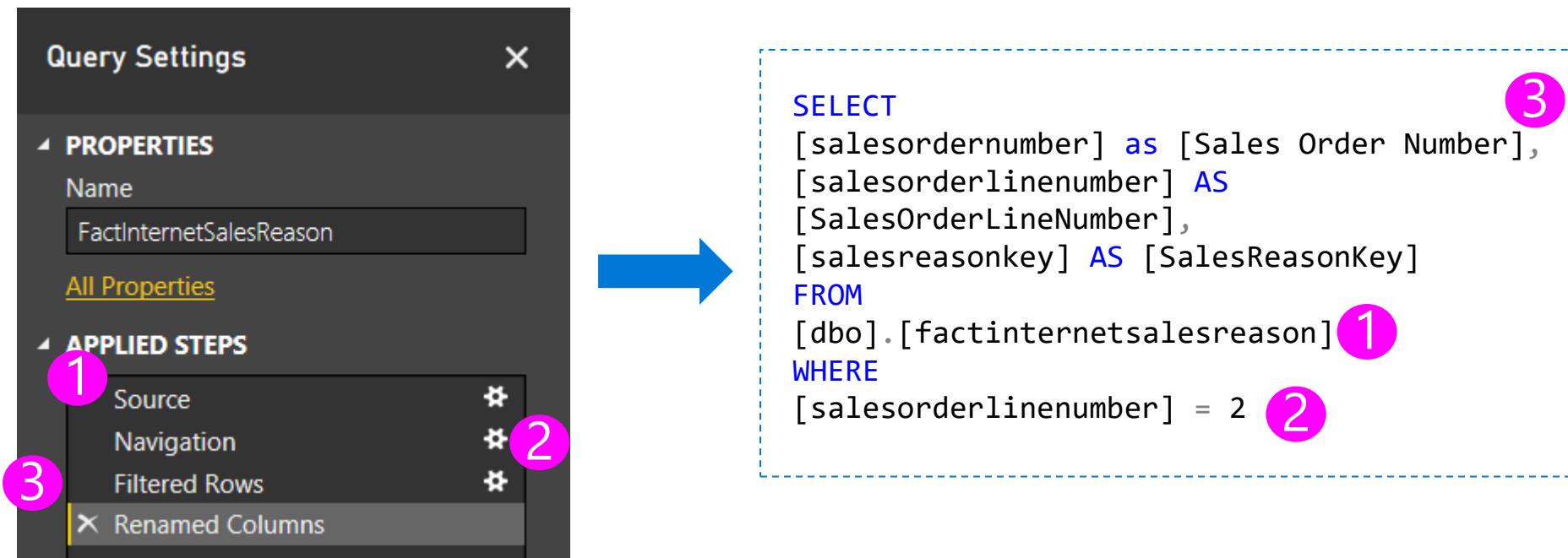
# Shaping Data

## Query folding

In some cases, Power BI Desktop will be able to **push the transformation work to the data source**

This is important from an **optimization perspective**.

**If query folding doesn't happen, all the data has to be loaded into the tool**



# Shaping Data

## Query folding

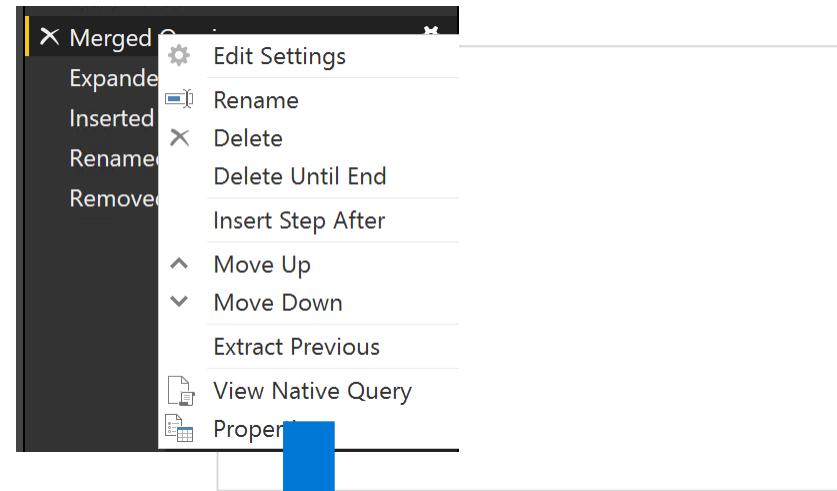
### Supported **data sources**:

- Relational Databases
- Odata
- Exchange
- Active Directory

Not all transformations are supported (for instance, custom SQL prevents folding)

In order to see what query is being generated use the “View Native Query” option, for a given step.

As the tool evolves, more transformations and sources will support folding



### Native Query

```
select [$Outer].[OrderDate] as [OrderDate],  
[$Outer].[Quantity] as [Quantity],  
[$Outer].[UnitPrice] as [UnitPrice],  
[$Outer].[DiscountAmount] as [Discount],  
[$Outer].[ProductSKU] as [Product SKU],  
[$Outer].[ProductName] as [Product Name],  
[$Outer].[ProductCategory] as [Product Category],  
[$Outer].[ItemGroup] as [Product Item Group],  
[$Outer].[Demographic] as [Product Demographic],  
[$Outer].[StateName] as [State Name],  
[$Inner].[RegionName] as [Region Name]  
from  
(  
    select [$Outer].[OrderNumber] as [OrderNumber],  
    [$Outer].[OrderDate] as [OrderDate],  
    ...  
)  
    ...  
) as [Inner]  
    ...  
) as [Outer]  
    ...  
) as [Main]  
    ...  
) as [Final]  
    ...  
) as [Result]
```

# Shaping Data

## Query formulas

The language was previously known as **M** or **Power Query Formula Language**

Has a standard library with built-in functions:

Is **case-sensitive** and **strongly typed**

Every transformation generates code automatically. Each step maps to an instruction.

Use the **Advanced Editor** to edit the query or **Blank Query** data source to start from an empty query

# Shaping Data

## Query formulas – an example

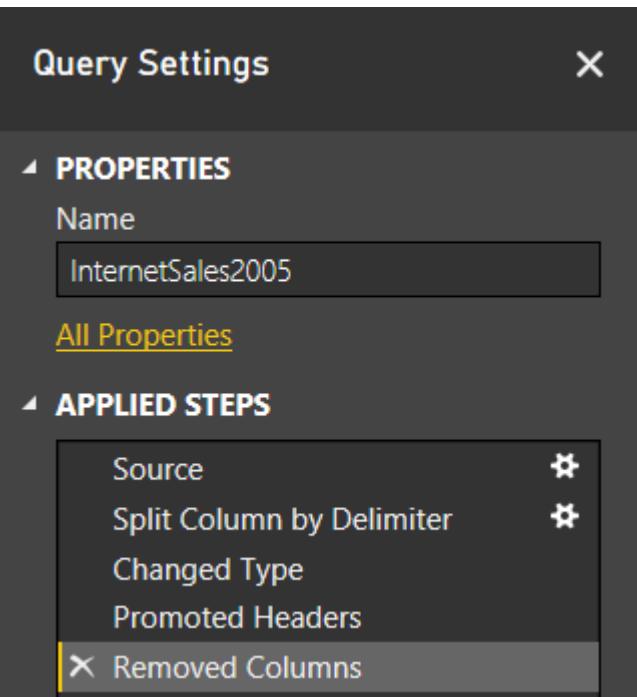
Every query starts with a **let**  
instruction and ends with an  
**in**

Each step name is prefixed with a # if it has spaces and its **name matches** exactly the one in the **Query Settings Pane** and the **order** is also the **same**

Several built-in functions implement the interface functionality with the **Object.Function(...)** syntax like `Table.RemoveColumns`

```
let Source = Table.FromColumns({Lines.FromBinary})
#"Split Column by Delimiter" = Table.SplitColumn(
    Source, "Text", Splitter.SplitTextByDelimiter("#"))
#"Changed Type" = Table.TransformColumnTypes(
    Source, {{"Text", typeText}})
#"Promoted Headers" = Table.PromoteHeaders(
    Source, [Header = "Text"])
#"Removed Columns" = Table.RemoveColumns(
    Source, {"Text"})

in
    #"Removed Columns"
```



```
etSales2005.txt"), null, null, 1252)),  
yDelimiter(", ", QuoteStyle.Csv), {"Colur  
1", type text}, {"Column1.2", type text  
"Column29", "Column30", "Column31", "(
```

# Shaping Data

## Query formulas – User Functions

Functions allow for **code reuse**. They require following syntax:

The procedure must be enclosed in a new **let** and **in** block

**Parameters** are declared in the function definition and then they can be used to make the procedure generic:

```
let
ParseInternetSalesFiles=(folder as text,file as text)=>
let
    Source = Table.FromColumns({Lines.FromBinary(File.Contents(folder & file),null,null,1252)}),
    #"Split Column by Delimiter" = Table.SplitColumn(Source,"Column1",Splitter.SplitTextByDelimiter(", ", QuoteStyle.Csv),{""
    #"Changed Type" = Table.TransformColumnTypes(#"Split Column by Delimiter",{{"Column1.1", type text}, {"Column1.2", type
    #"Promoted Headers" = Table.PromoteHeaders(#"Changed Type"),
    #"Removed Columns" = Table.RemoveColumns(#"Promoted Headers",{"Column27", "Column28", "Column29", "Column30", "Column31
in
    #"Removed Columns"
in
ParseInternetSalesFiles
```

```
let
ParseInternetSalesFiles=(folder as text file as text)=>
```

```
ry(File.Contents(folder & file),nul
Column(Source,"Column1",Splitter.S
    "Split Column by Delimiter",{""
    #"Changed Type" = Table.TransformColumnTypes(#"Split Column by Delimiter",{{"Column1.1", type text}, {"Column1.2", type
    #"Promoted Headers" = Table.PromoteHeaders(#"Changed Type"),
    #"Removed Columns" = Table.RemoveColumns(#"Promoted Headers",{"Column27", "Column28", "Column29", "Column30", "Column31
in
    #"Removed Columns"
in
ParseInternetSalesFiles
```

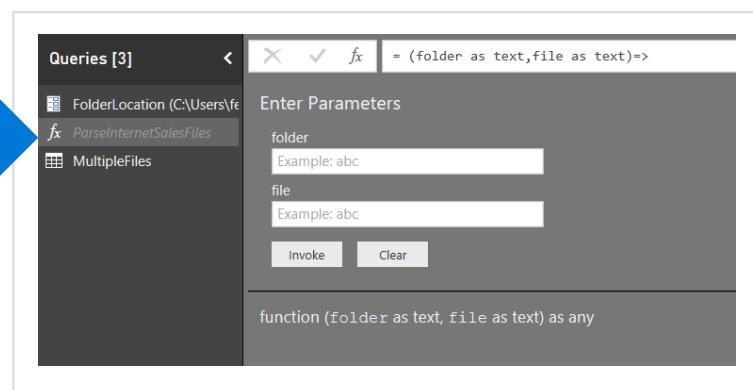
# Shaping Data

## Query formulas – User Functions

You can **invoke** them via the Query Editor

Notice the **fx** icon in the Left Pane that indicates a user function

Supports parameters of **type** “**Table**”



Queries [3] = (folder as text,file as text)>

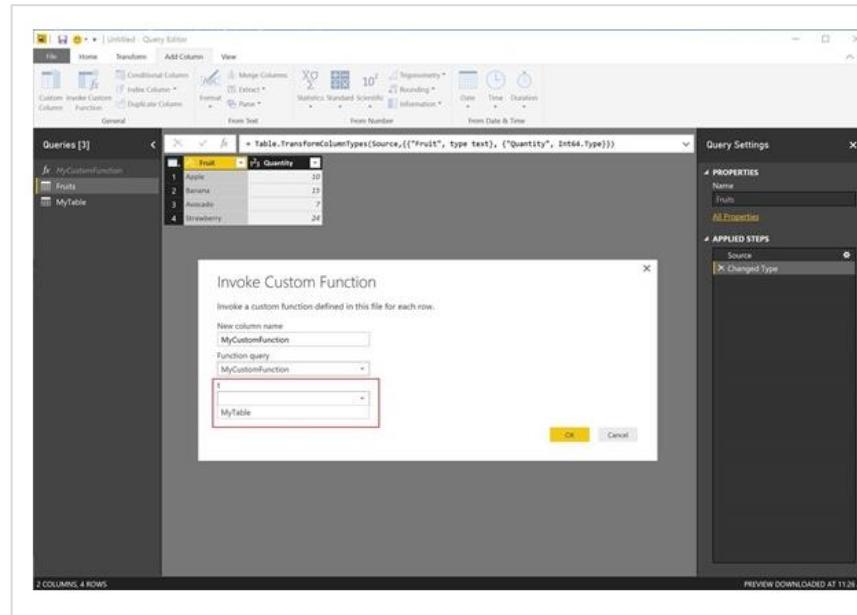
Enter Parameters

folder  
Example: abc

file  
Example: abc

Invoke Clear

function (folder as text, file as text) as any



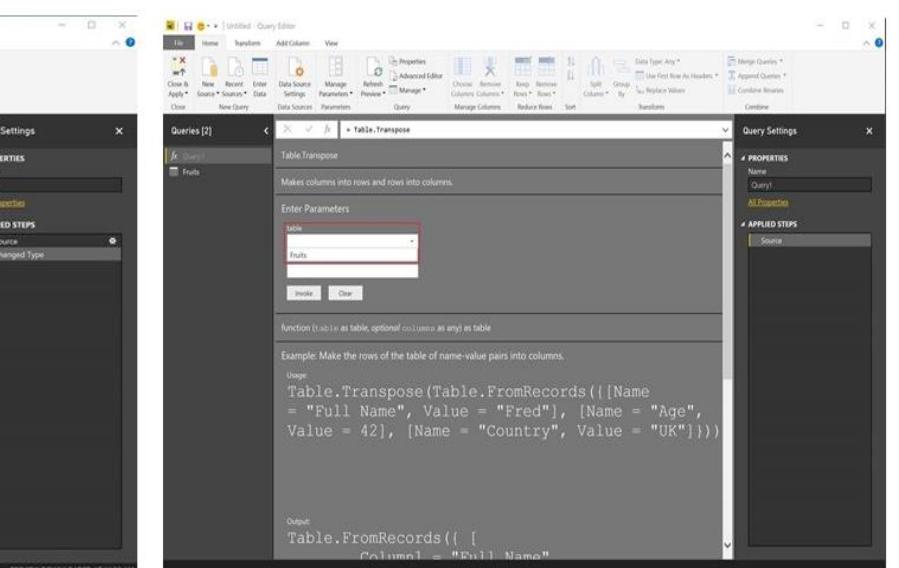
Queries [3] > Table.TransformColumnTypes(Source,{{"Fruit", type text}, {"Quantity", Int64.type}})

Invoke Custom Function

New column name: MyCustomFunction

Function query: MyCustomFunction

OK Cancel



Queries [2] > Table.Transpose

TableTranspose

Makes columns into rows and rows into columns.

Enter Parameters

Table  
Fruit

Invoke Clear

Function (Table) as table, optional columns as any table

Example: Make the rows of the table of name-value pairs into columns.

Usage:

Table.Transpose(Table.FromRecords({[Name = "Full Name", Value = "Fred"], [Name = "Age", Value = 42], [Name = "Country", Value = "UK"]}))

Output:

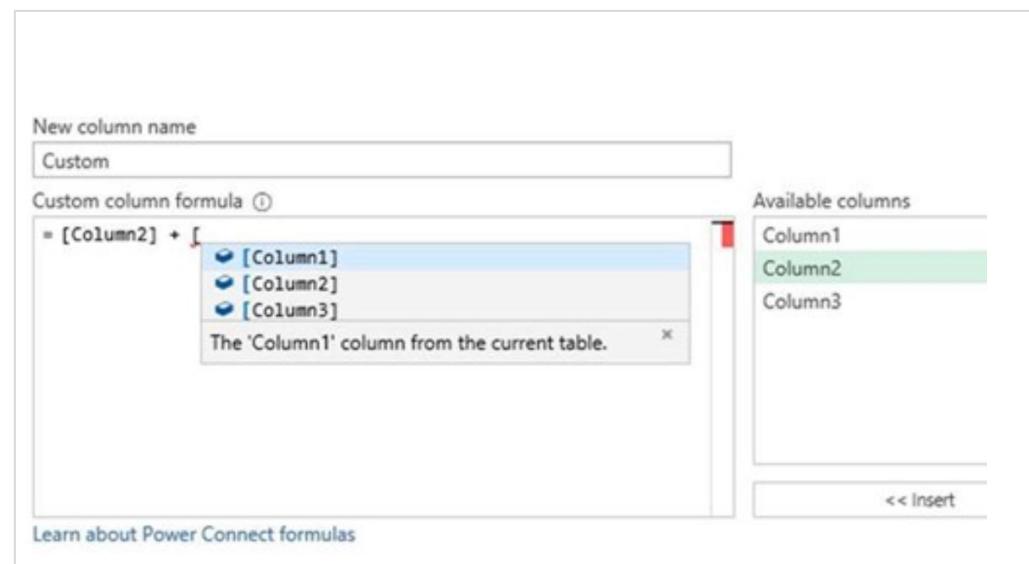
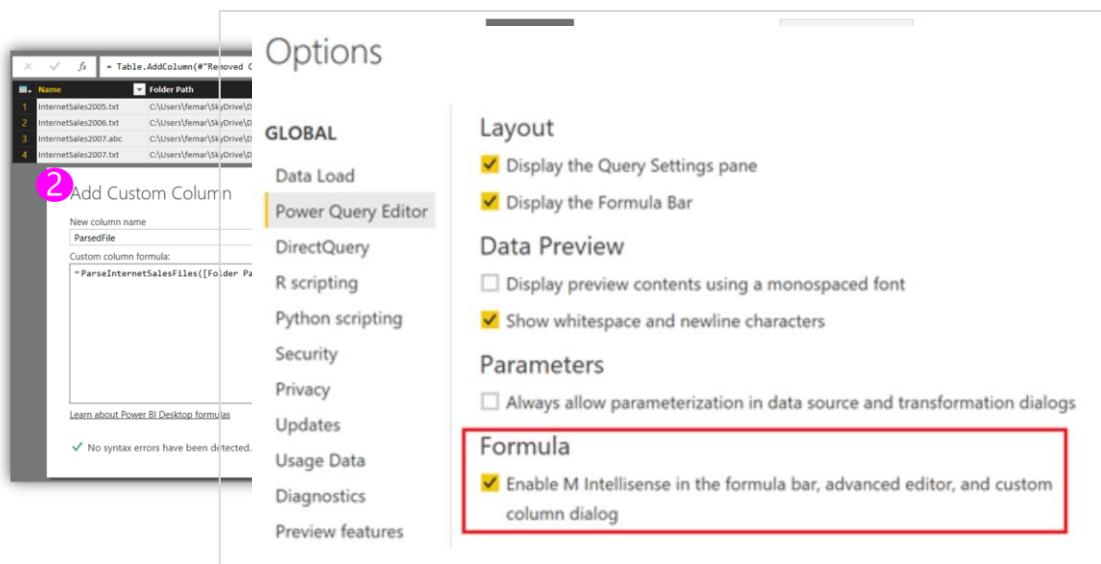
Table.FromRecords([  
Column = "Full Name",

# Shaping Data

## Query formulas – User Functions

You can also invoke them with the **Add Custom Column** task or with **Invoke Custom Function**.

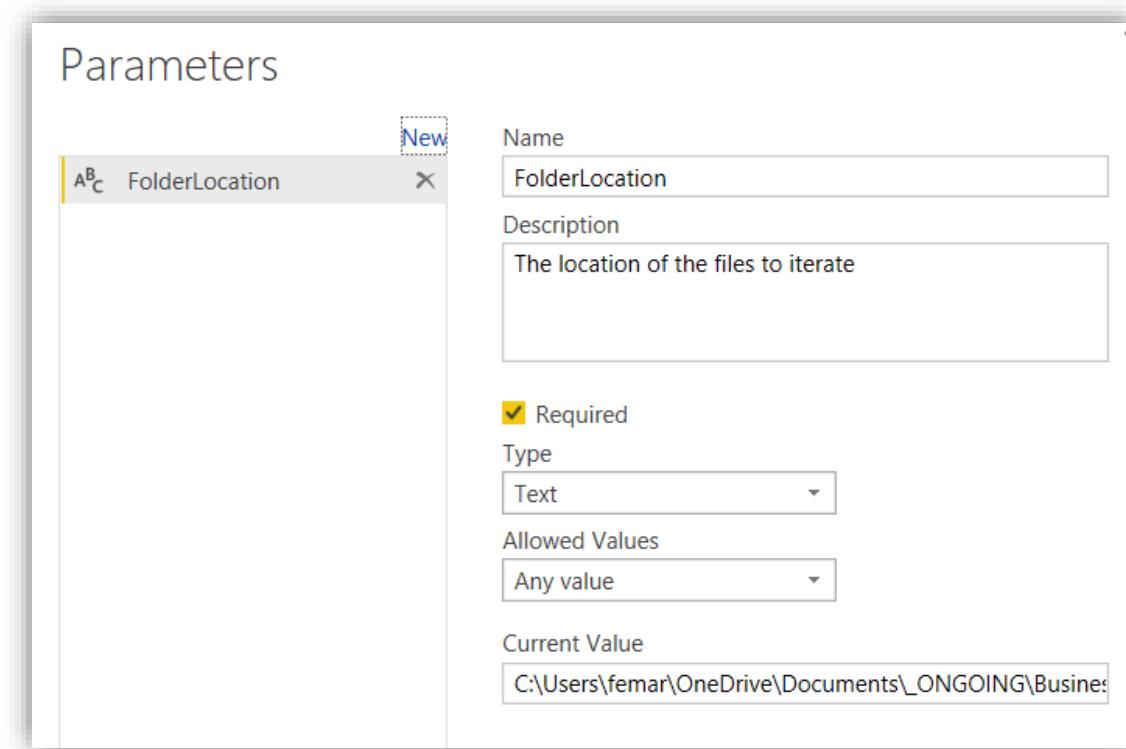
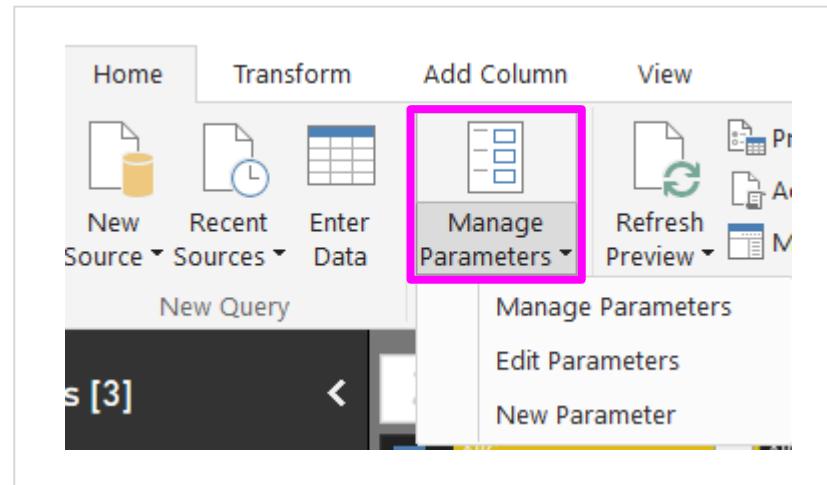
M Intellisense is supported. Can be turned off in **options**.



# Shaping Data

## Query Parameters

- Queries and other objects can use parameters to get a **dynamic behavior**



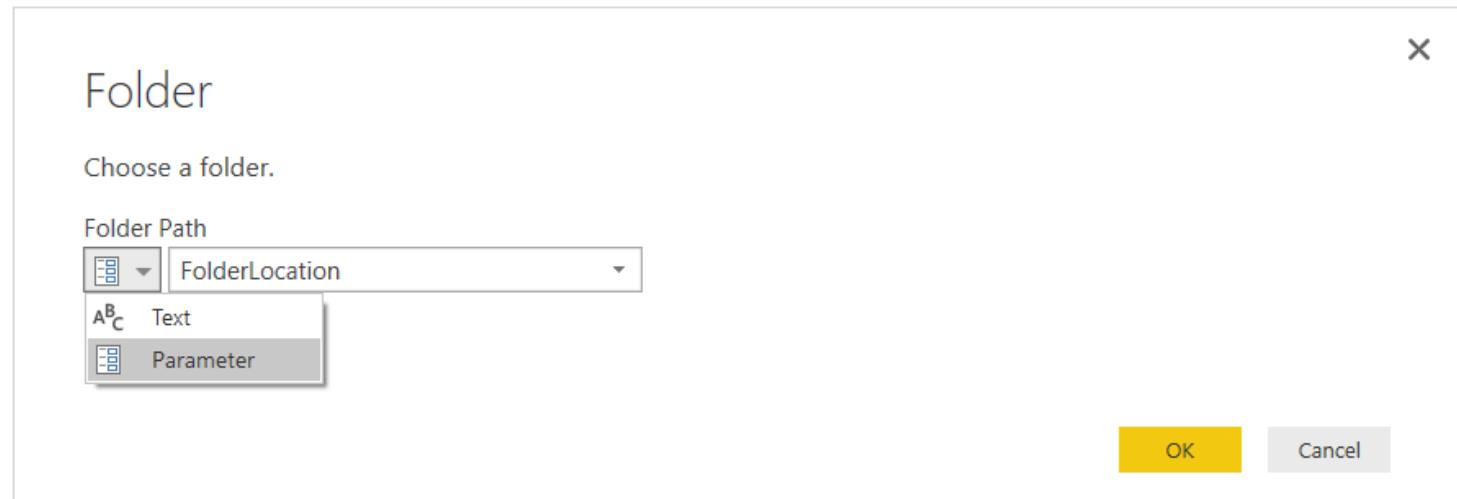
# Shaping Data

## Query Parameters

Each parameter generates a **query**

Can be referenced from other queries and loaded to the data model for use in calculations

Support for parameters is available throughout the tool



# Shaping Data

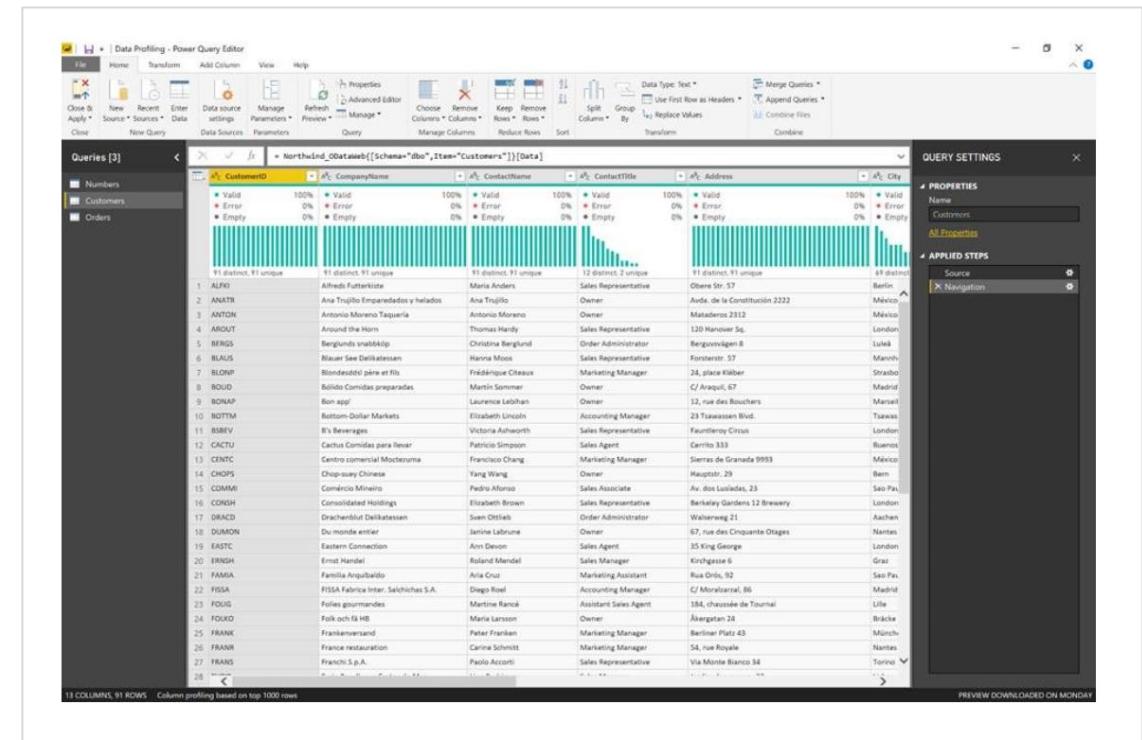
## Query data profiling

Allowing you to easily identify **errors** and **empty values** in your data previews

**Quality bar** below the column headers indicating whether there were any **error values** found or not.

“**Column quality**” option under the View tab

**Column Distribution** allows you to get a **sense** for the **overall distribution** of values within a column in your data previews, including the **count of distinct** values and **unique** values



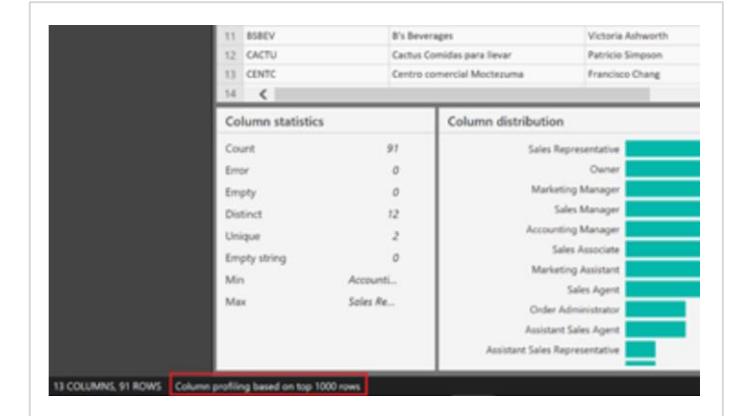
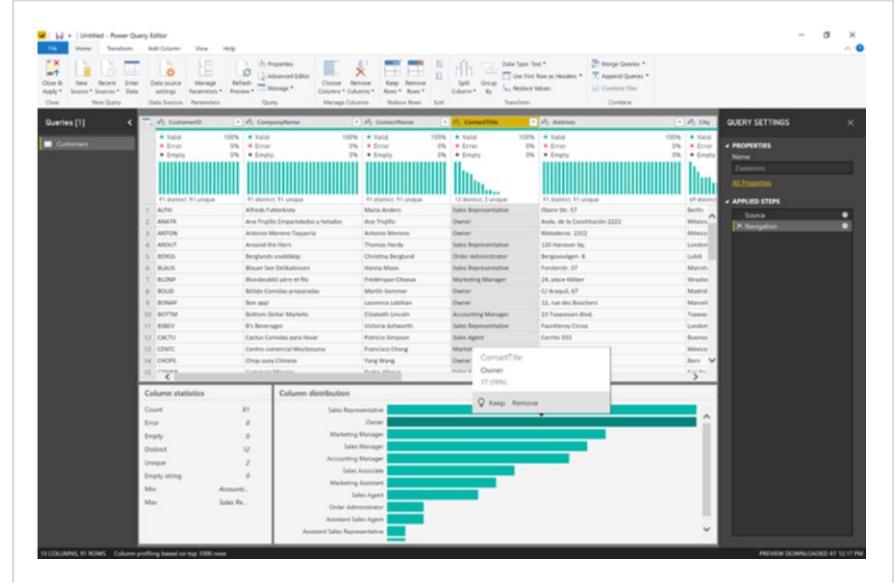
# Shaping Data

Query data profiling

**Column Profiles Pane** – to provide deeper profiling capabilities for any column. Such as

- **Column Statistics** - # of errors, empty, valid, duplicated and unique values. Value distribution measures such as Min/Max/Average/Median, etc.
- **Column distribution** – Larger size version of the inline value distribution histograms, also including the ability to Keep or Remove values, which will generate the corresponding Filter Rows step in your query (“Equals” / “Does Not Equal” filters).

You can also now switch statistics based on **preview** to the entire table.



# Shaping Data

## Query data profiling

- **Visibility** of each of the data profiling elements can be controlled through the **View** tab.

The screenshot shows the Power Query Editor interface with the 'View' tab selected. In the 'Layout' section of the ribbon, three checkboxes are highlighted with red boxes: 'Column distribution', 'Column profile', and 'Column quality'. The 'Column distribution' checkbox is checked. Below the ribbon, the 'Data Preview' pane displays a table with three columns: 'CustomerID', 'CompanyName', and 'ContactName'. Each column has a summary row above the data rows, showing counts for 'Valid', 'Error', and 'Empty' values, and a note below stating '91 distinct, 91 unique'. The table contains two data rows: 'ALFKI' and 'ANATR' under CustomerID, 'Alfreds Futterkiste' and 'Ana Trujillo Emparedados y helados' under CompanyName, and 'Maria Anders' and 'Ana Trujillo' under ContactName.

	CustomerID	CompanyName	ContactName
1	ALFKI	Alfreds Futterkiste	Maria Anders
2	ANATR	Ana Trujillo Emparedados y helados	Ana Trujillo

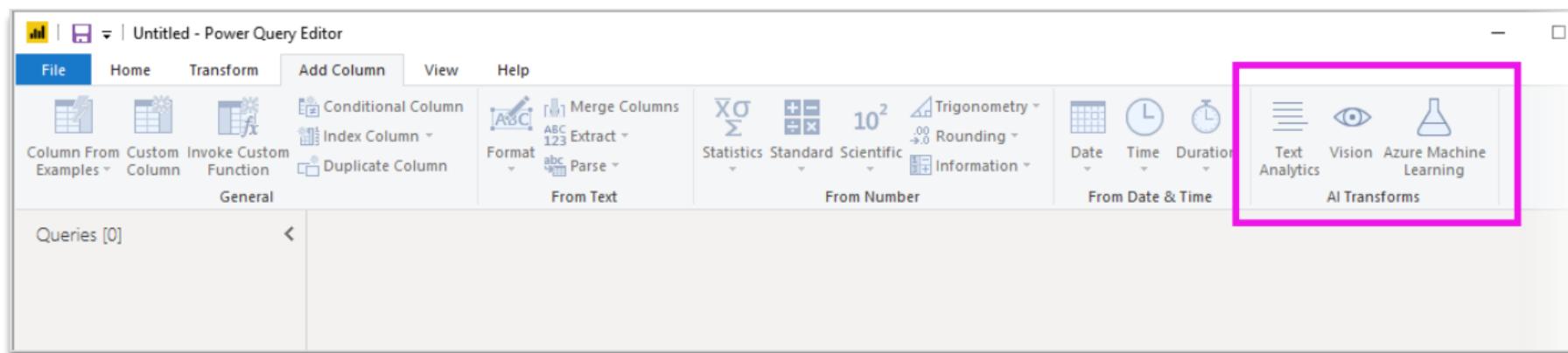
# Shaping Data

## AI Insights Query Editor

Use **AI Insights** to gain access to a collection of pre-trained machine learning models that enhance your data preparation efforts.

With **Text Analytics and Vision in Power BI**, you can apply different algorithms from Azure Cognitive Services to enrich your data in Power Query.

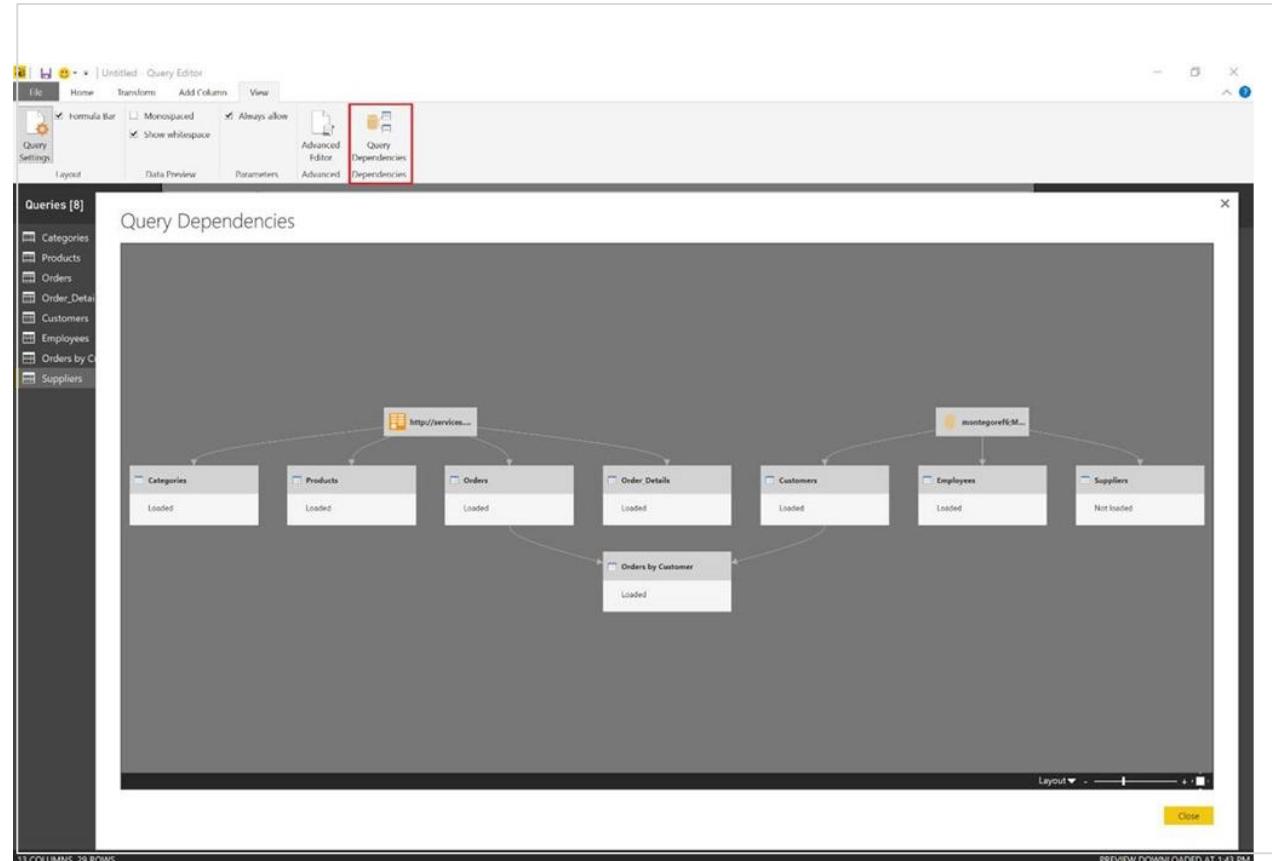
Power BI makes it simple to incorporate the insights from **models hosted on the Azure Machine Learning service**, using straightforward point-and-click gestures



# Shaping Data

## Query dependency view

- **View dependencies across all queries and data sources**



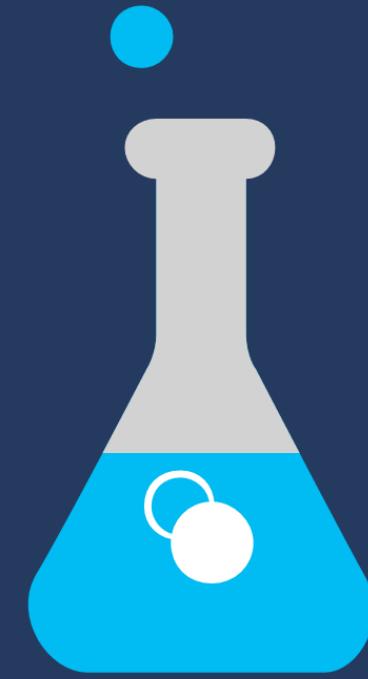
# Demonstration

- Power Query Editor walkthrough



# Power BI Desktop

- Shaping Data



LAB

**Questions?**



# Knowledge Check

Why do we need to shape our data?

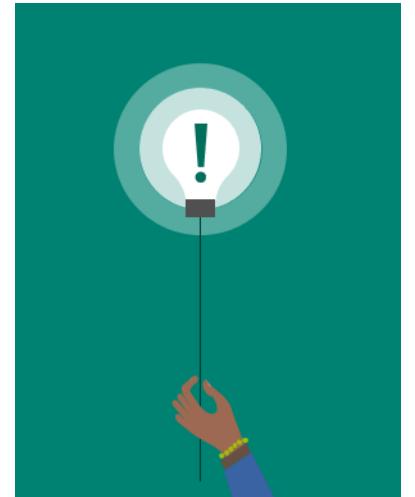
What are some examples of available data transformations in Power BI Desktop?

# Lesson 4: Creating a Data Model

# Objectives

After completing this learning, you will be able to:

- Describe modeling data to meet business requirements
- Learn different relationship models
- Learn different data types



# Creating a Data Model

---

After data connections are created and the data is shaped according to the business requirements, we start modeling it

---

Relationships between the extracted tables can be established to allow filtering

---

Calculations can be created for additional context or for implementing business metrics or even key performance indicators

---

Data can be categorized, typed, and formatted

---

Custom sorting can be implemented for the attributes

# Creating a Data Model

## Data View

The screenshot shows the Power BI Data View interface. At the top, there's a ribbon with tabs like File, Home, Help, External Tools, and Table tools (which is currently selected). Below the ribbon, there's a search bar labeled 'Name' with 'Sales' typed in. To the right of the search bar are icons for 'Mark as date table', 'Calendars', 'Manage relationships', 'Relationships', 'New measure', 'Quick measure', 'New column', and 'New table'. A 'Calculations' section follows. On the left, there's a sidebar with a 'Structure' tab showing a hierarchical tree view of the data model. The main area is a 'Data Grid' (labeled A) displaying a table with columns: OrderDate, Quantity, UnitPrice, Discount, Product SKU, and Product Name. The data shows multiple rows of sales records from April 5 to April 10, 2014. A context menu is open over one of the rows, with 'Sort descending' highlighted (labeled B). A pink box labeled 'Filtering & sorting options' highlights the filter context menu and the 'Date filters' dropdown. The 'Fields' pane (labeled C) on the right lists various tables and columns: CostReduction, CostReduction-WhatIf, Date, Manager, Product, and ProductCost. A search bar is also present in the Fields pane. A formula bar at the bottom (labeled D) contains DAX code: `Sort ascending`, `Sort descending` (with a checked checkbox), `Clear sort`, `Clear filter`, `Clear all filters`, `Date filters` (highlighted with a pink box), and a 'Search' dropdown. The 'Date filters' dropdown (labeled E) shows several filter options: '(Select all)', '10/17/2011', '10/19/2011', and '10/20/2011'. A pink box highlights the '10/17/2011' option.

A Data View icon

B Data Grid – Shows the data for a selected table

C Modeling Ribbon – Manage relationships, calculations, data types, formats, and categorization

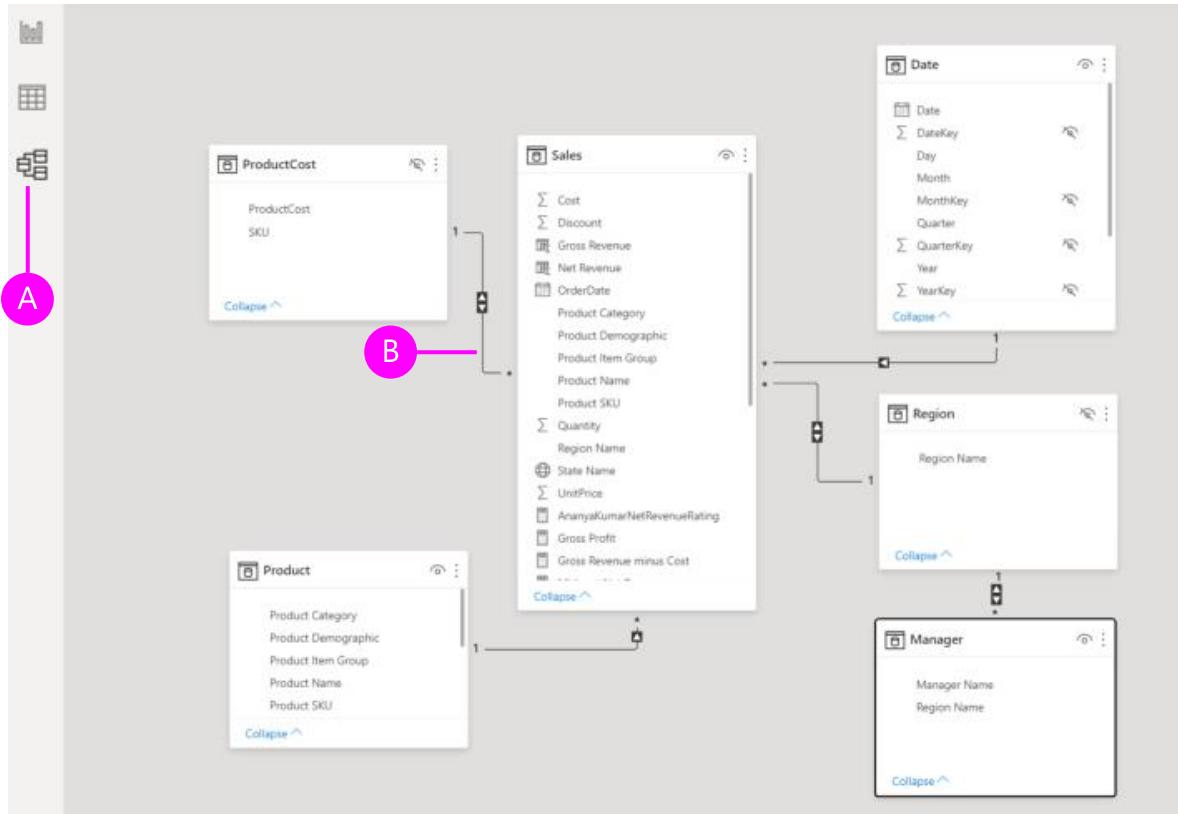
D Formula bar – DAX formulas for calculations

E Search – Search for tables or column names

F Fields List – Select a table or column to view in the Data Grid

# Creating a Data Model

## Model View



**A** Model view icon – shows the relationships in the model. Here you can create relationships or view them.

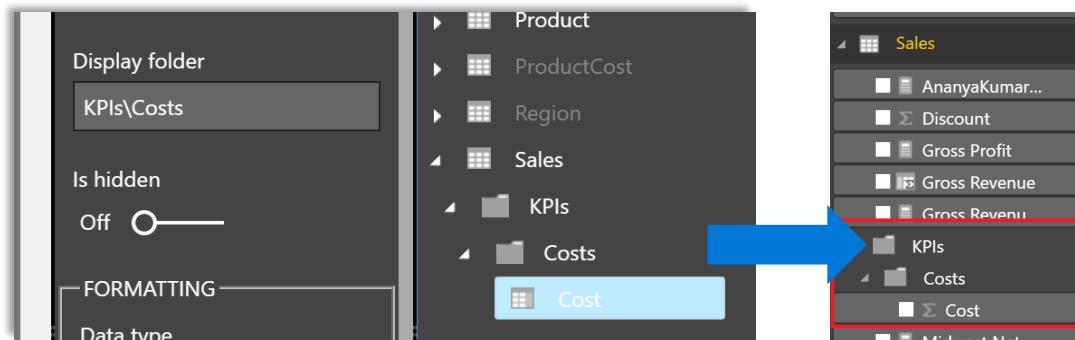
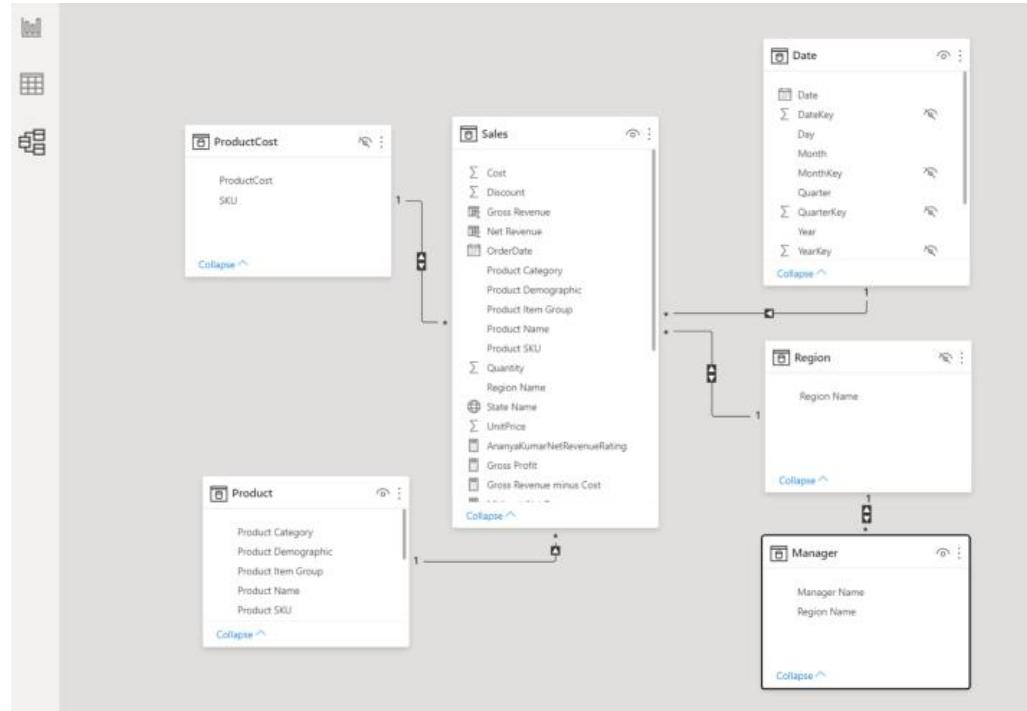
**B** Relationship – you can hover your cursor over a relationship to show the columns used.

Double-click on a relationship to open it in the **Edit Relationship** dialog box

In the example, Sales and Stores have a relationship via StoreKey

# Creating a Data Model

## Model View



Allows the creation of **customized layouts (subsets)**

Multiple diagram layouts

**Bulk** editing of columns, measures, and tables.

Allows setting object (**tables or columns**) properties directly on the diagram

Allows setting properties *en masse*

Allows the creation of display folders

# Creating a Data Model

## Relationships

They **connect two tables** allowing filtering and calculations using columns from both. **No need to flatten!**

They are based on a **single column** from **each table**. One table acts as a lookup table and the other as a referencing table

The **datatypes** for the columns **do not have to be the same**

Relationships can be **created manually** or **automatically inferred** by the tool

**More than one relationship can exist** between two tables

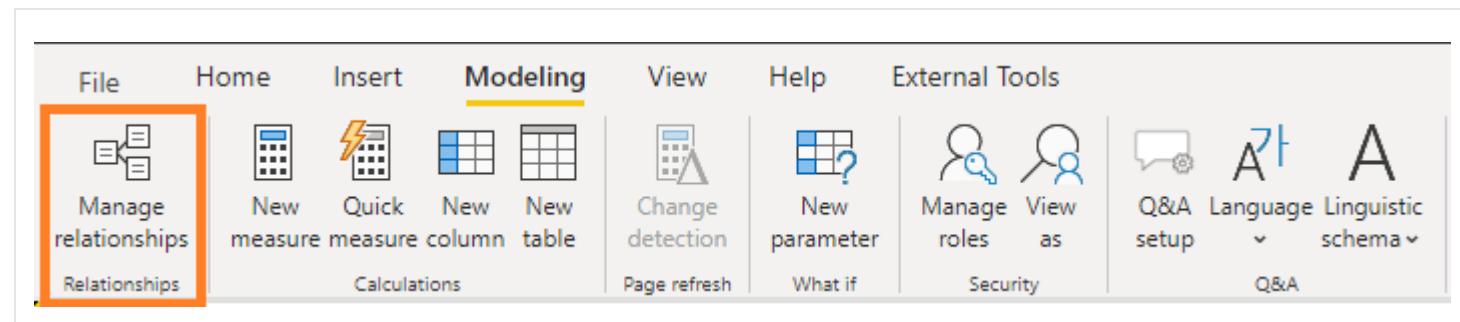
# Creating a Data Model

## Relationships – Creation

Relationships can be created manually by using the **Manage Relationships** tab.

Or by dragging-and-dropping columns in the Modelling View

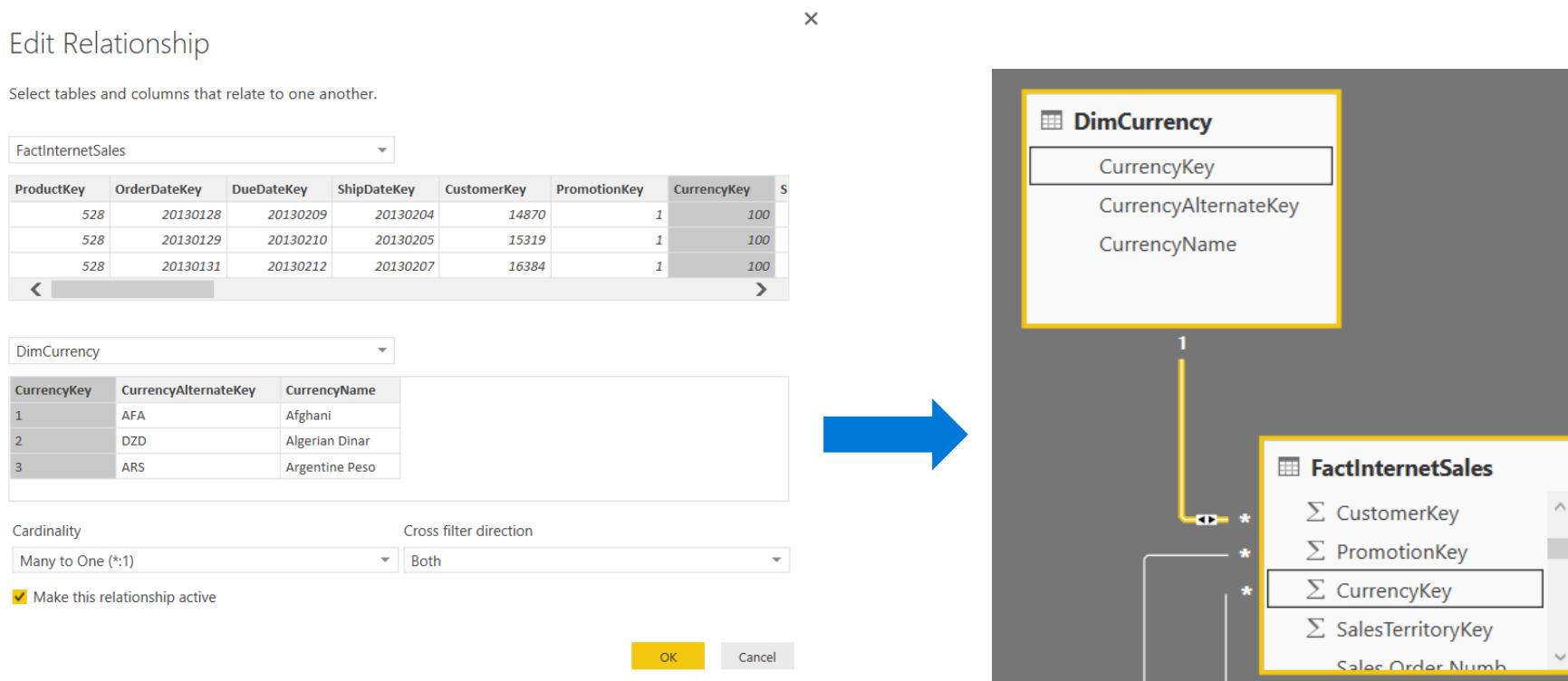
There is also an **auto-detect** feature where Power BI desktop will rely on **column names and data types** (or constraints defined in the data source) to detect relationships



# Creating a Data Model

## Relationships - Cardinality

- Many to One (\*:1) or One to Many (1:\*)



# Creating a Data Model

## Relationships – One to One (1:1)

Create Relationship

Select tables and columns that relate to one another.

ProjectBudget

Approved Projects	BudgetAllocation	AllocationDate
Blue	40000	Saturday, December 1, 2012
Red	100000	Saturday, December 1, 2012
Green	50000	Saturday, December 1, 2012

CompanyProjectPriority

Project	Priority
Blue	A
Red	B
Green	C
Yellow	C
Purple	B

Advanced options

Cardinality

One to One (1:1)

Cross filter direction

Both

Make this relationship active

In a One to One relationship **there are no repeating values** for the column in either table.

Power BI Desktop will **automatically detect** this and set it automatically.

Only set it if **always true**



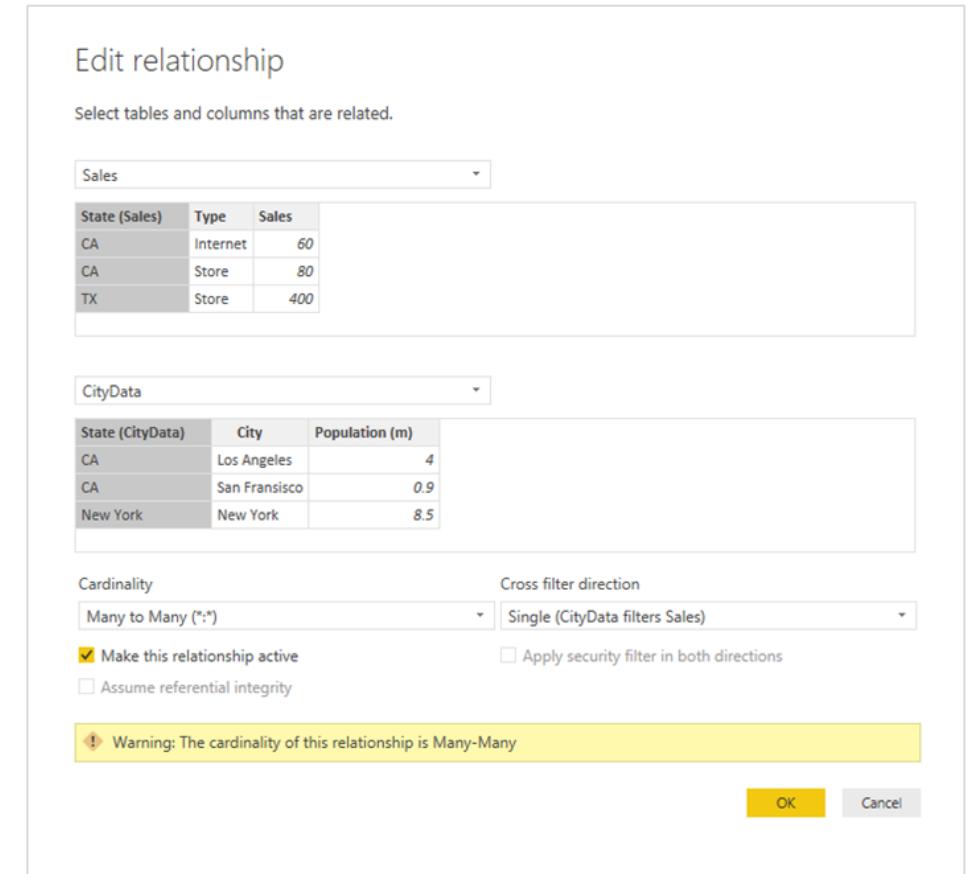
Project	Priority	BudgetAllocation	AllocationDate
Blue	A	40,000	12/1/2012
Red	B	100,000	12/1/2012
Green	C	50,000	12/1/2012
Yellow	C		
Purple	B		
Orange	C		

# Creating a Data Model

Relationships – Many to Many

This relationship indicates that neither of the tables contain unique values.

Power BI Desktop will **automatically detect** this and display a warning.



# Creating a Data Model

## Relationships - Direction

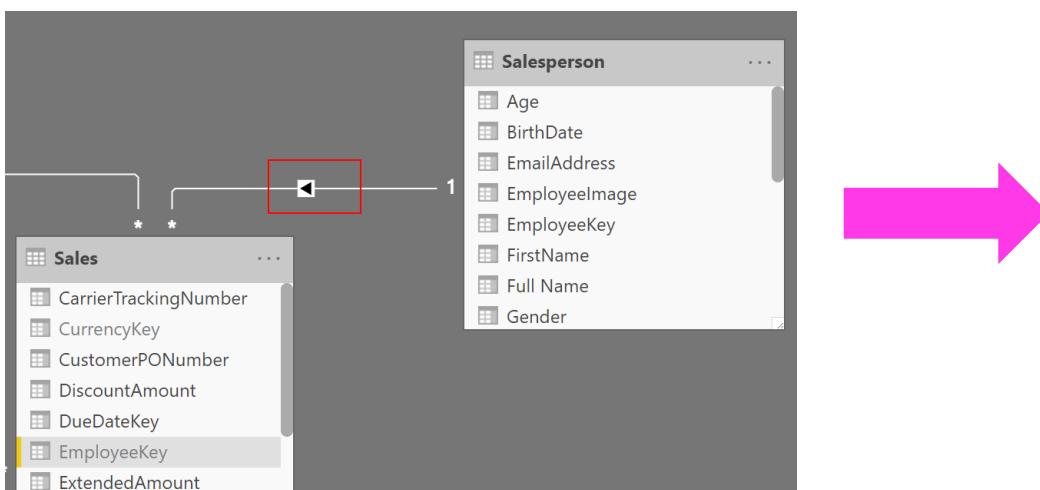
The direction of the relationship determines how data can be filtered.

**Single** is the typical behavior in star-schemas where dimension tables (lookup) filter fact tables (transactional tables). **filter propagates from one side to the many side** of the relationship (Dimension to Fact table). This allows for slicing and filtering fact data using dimension attributes

**Both** tables act as if together in a single table, so filtering is possible on either side of the relationship.

Cross filter direction

Both  
Single  
Both

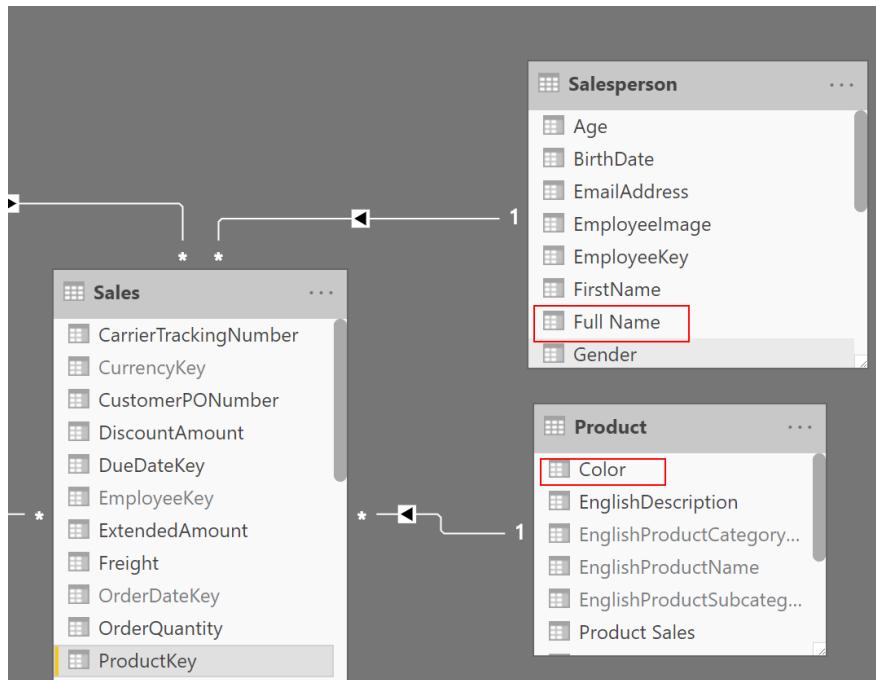


Full Name	SalesAmount
Abbas, E Syed	\$172,524.45
Alberts, E Amy	\$732,078.44
Anzman-Wolfe, O Pamela	\$3,325,102.60
Blythe, G Michael	\$9,293,903.01
Campbell, R David	\$3,729,945.35
Carson, Jillian	\$10,065,803.54
Ito, K Shu	\$6,427,005.56
Jiang, Y Stephen	\$1,092,123.86
Mensa-Annan, A Tete	\$2,312,545.69
Mitchell, C Linda	\$10,367,007.43
Pak, B Jae	\$8,503,338.65
Reiter, Michael Tsvi	\$7,171,012.75
Saraiva, Edvaldo José	\$5,926,418.36
Tsoflias, N Lynn	\$1,421,810.93
Valdez, B Rachel	\$1,790,640.23
Vargas, R Garrett	\$3,609,447.22
Varkey Chudukatil, R Ranjit	\$4,509,888.93
<b>Total</b>	<b>\$80,450,596.98</b>

# Creating a Data Model

## Relationships – Direction

- In some cases, we may want filter to propagate from many to the one side of the relationship.
- E.g. Show number of products of each color sold by each salesperson.
- **Wrong result using “Single” direction**

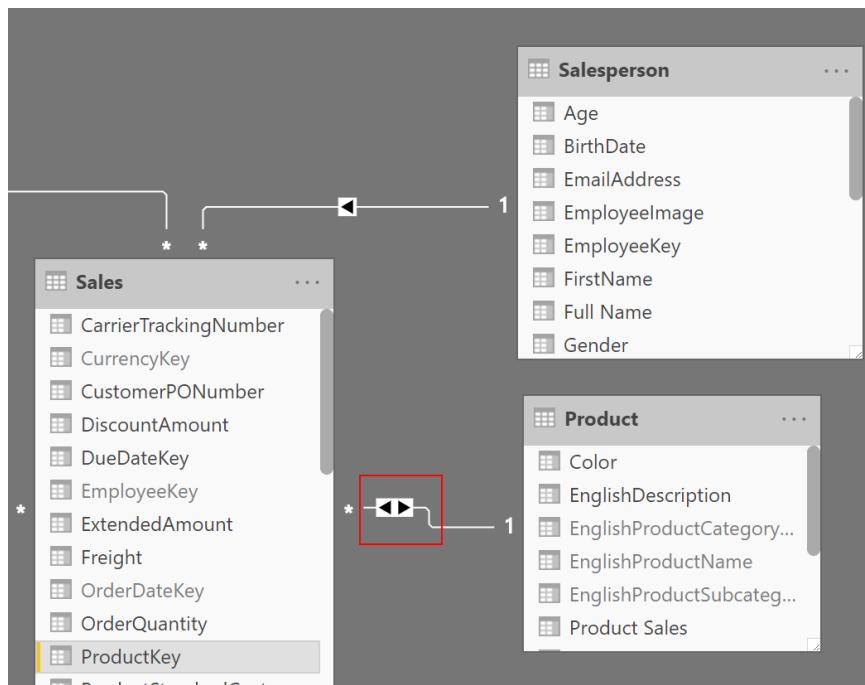


Full Name	Color	Count of Color
Abbas, E Syed	Black	129
	Blue	28
	Multi	18
	NA	56
	Red	63
	Silver	45
	Silver/Black	7
	Yellow	46
	<b>Total</b>	<b>397</b>
Alberts, E Amy	Black	129
	Blue	28
	Multi	18
	NA	56
	Red	63
	Silver	45
	Silver/Black	7
	White	4
	Yellow	46
	<b>Total</b>	<b>397</b>

# Creating a Data Model

## Relationships – Direction

- In some cases, we may want filter to propagate from many to the one side of the relationship.
- E.g. Show number of products of each color sold by each salesperson.
- **Correct result using “Both” direction**

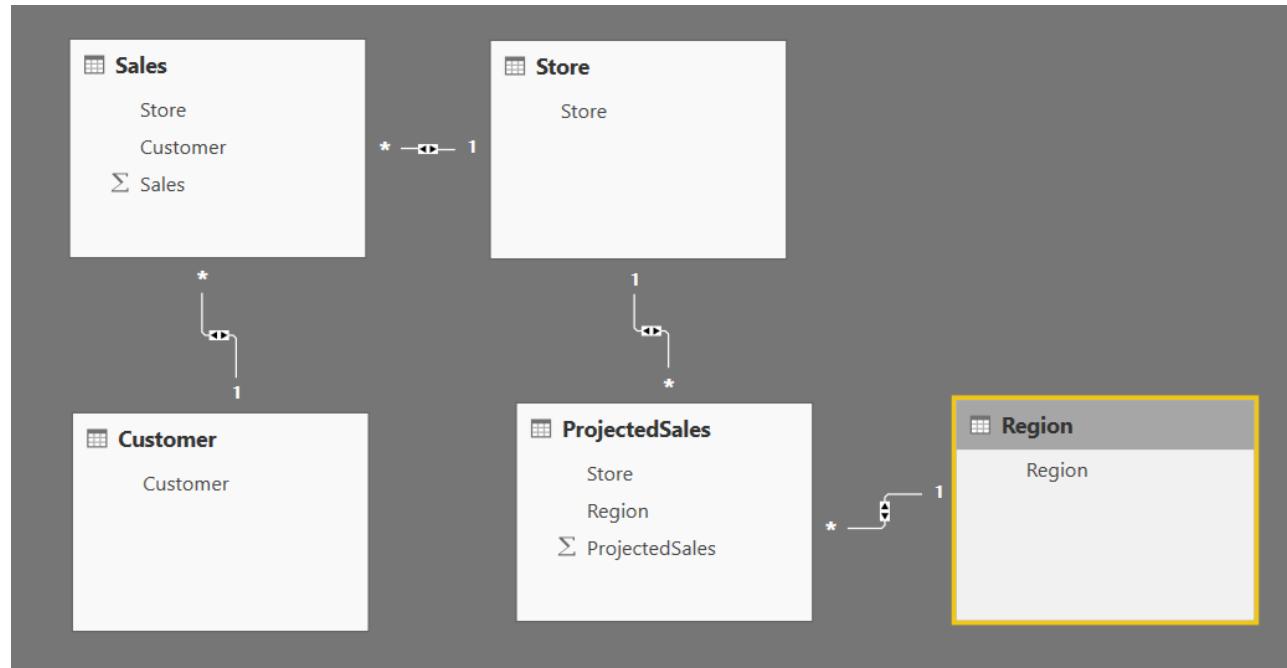


Full Name	Color	Count of Color
Abbas, E Syed	Black	10
	Blue	19
	Multi	4
	NA	10
	Red	1
	Silver	6
	Silver/Black	2
	Yellow	16
	<b>Total</b>	<b>68</b>
Alberts, E Amy	Black	76
	Blue	22
	Multi	11
	NA	27
	Red	18
	Silver	32
	Silver/Black	7
	White	2
	Yellow	38
	<b>Total</b>	<b>233</b>

# Creating a Data Model

## Relationships – Direction

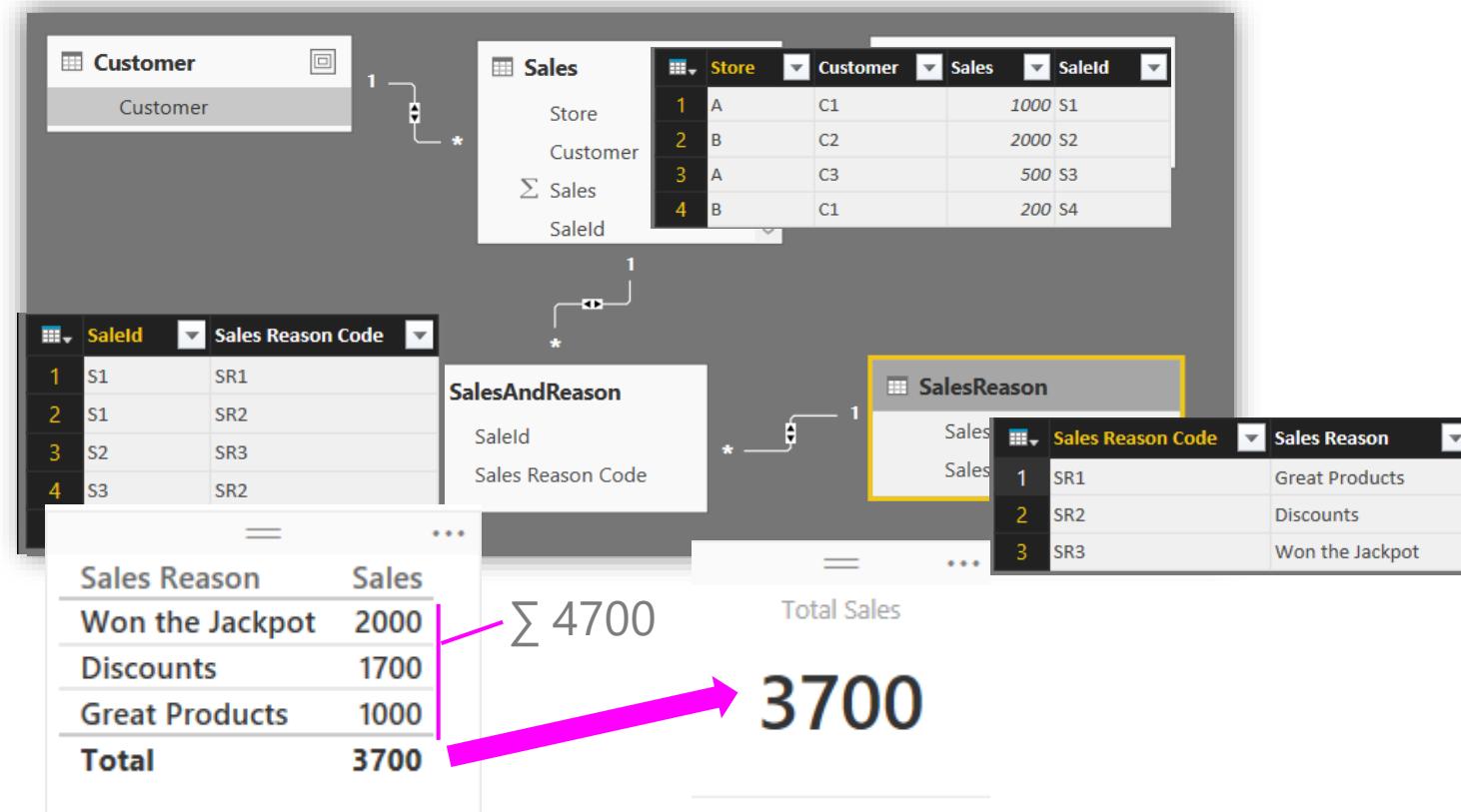
- Don't turn on "Both" direction for every relation. It is **slower** and might return **unexpected results**



# Creating a Data Model

## Relationships – Direction

The Both direction type also enables Many to Many relationships



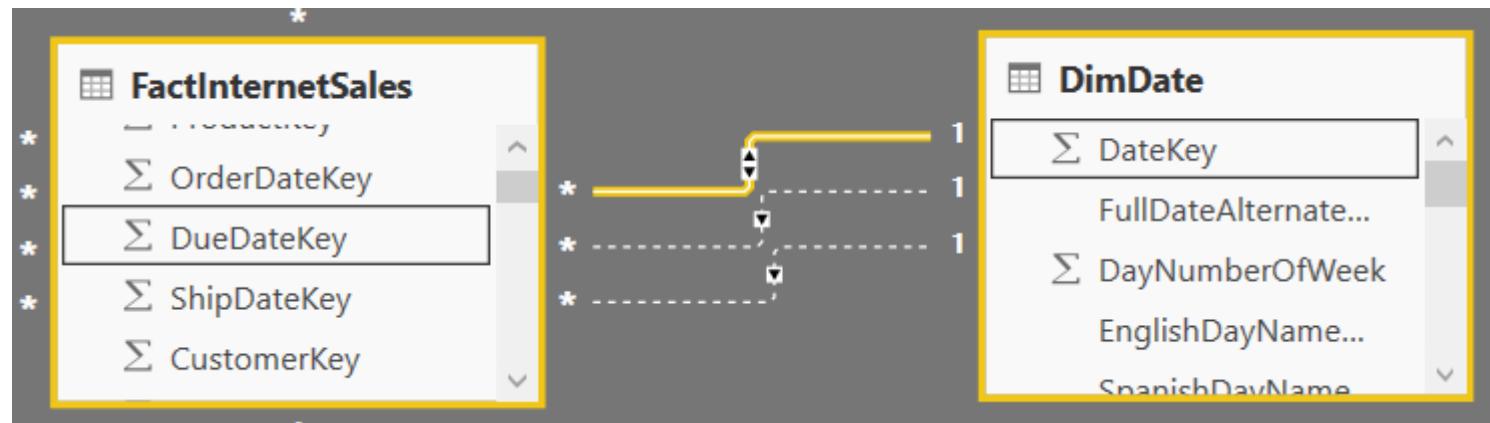
# Creating a Data Model

Relationships – Active and Inactive

**More than one** relationship can exist between two tables, but **only one can be active** at a time

The **columns** from the transactional table must be **different**

In the above example, three relationships exist. The **full line** represents the **active** (in this case, we aggregate on Due Date)

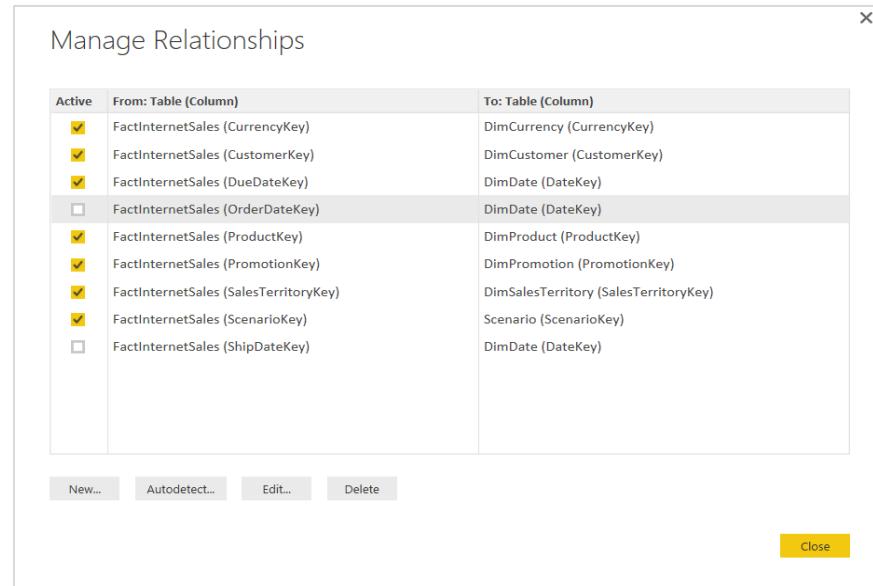


# Creating a Data Model

## Relationships – Active and Inactive

It is possible to **control which relationship is active** by using **the Manage Relationships window**

It is also possible to use **DAX to activate a relationship** in a calculation or **import the lookup table more than once**



# Creating a Data Model

## Relationships – Parent-Child

- Not supported natively. As a workaround, **denormalize** the table.

The diagram illustrates the transformation of a Parent-Child relationship into a denormalized table structure. On the left, a blue-bordered table titled "Parent-Child" contains three rows of data. The first row has "Key" in the first column and "Parent" in the second. The second row has "Lisbon" in the first column and "Portugal" in the second. The third row has "Europe" in the first column and "Europe" in the second. A large blue arrow points from this table to the right. On the right, another blue-bordered table titled "Denormalized" contains four columns: "Key", "District", "Country", and "Region". The same three rows of data are present: "Sintra" in the "Key" column, "Lisbon" in the "District" column, "Portugal" in the "Country" column, and "Europe" in the "Region" column.

Parent-Child	
Key	Parent
Sintra	Lisbon
Lisbon	Portugal
Portugal	Europe

Denormalized			
Key	District	Country	Region
Sintra	Lisbon	Portugal	Europe

# Creating a Data Model

## Relationships – Data Quality Issues

If data has **null** or **blank values** for the columns that support relationships either:

- Remove those rows (which might affect statistics and summarizations)
- Define a default value that replaces those values so that a match is made

If **duplicate values** exist in a lookup table, a relationship cannot be created

- Remove the duplicates so that the lookup table contains none. You can use the data shaping features of Power BI Desktop to do it.

# Creating a Data Model

## Relationships – Options

**Import relationships from data sources on first load** (like foreign keys in relational databases)

**Update or delete relationships when refreshing data**– might remove manual relationships

**Autodetect new relationships after data is loaded** – useful when adding more tables to the model

## Options

### GLOBAL

- Data Load
- Power Query Editor
- DirectQuery
- R scripting
- Python scripting
- Security
- Privacy
- Regional Settings
- Updates
- Usage Data
- Diagnostics
- Preview features
- Auto recovery
- Report settings

### CURRENT FILE

- Data Load
- Regional Settings
- Privacy
- Auto recovery
- DirectQuery
- Query reduction
- Report settings

### Type Detection

- Automatically detect column types and headers for unstructured sources

### Relationships

- Import relationships from data sources on first load ⓘ
- Update or delete relationships when refreshing data ⓘ
- Autodetect new relationships after data is loaded ⓘ

[Learn more](#)

### Time intelligence

- Auto date/time ⓘ

### Background Data

- Allow data preview to download in the background

### Parallel loading of tables

- Enable parallel loading of tables ⓘ

### Q&A

- Turn on Q&A to ask natural language questions about your data ⓘ

OK

Cancel

# Creating a Data Model

## Relationships – DirectQuery

**Assume Referential Integrity**  
is an optimization to tell the  
engine that there are **no**  
**“lookup failures”** between the  
two tables – typical in data  
warehousing scenarios between  
facts and dimensions

### Edit relationship

Select tables and columns that are related.

Sales

OrderNumber	OrderDate	ShipDate	CustomerStateID	ProductID	Quantity
TT00017434	Tuesday, January 1, 2013	Wednesday, January 9, 2013		14	3
TT00017445	Wednesday, January 2, 2013	Friday, January 4, 2013		13	3
TT00017456	Sunday, January 6, 2013	Wednesday, January 9, 2013		35	3

Product

ProductID	ProductSKU	ProductName	ProductCategory	ItemGroup	KitType	Channels	Demo
1	1010-GL120-3C	Trainer - Tailspin GL-120	Glider	Airplane	RTF	3	Novice
2	1010-GL155-4C	Trainer - Tailspin GL-155	Glider	Airplane	RTF	4	Intermediate
3	2030-PCUB-3C	Piper Cub 3 Channel	Trainer	Airplane	RTF	3	Beginner

Cardinality

Many to one (\*:1)

Cross filter direction

Both

Make this relationship active

Apply security filter in both directions

Assume referential integrity

[Learn more](#)

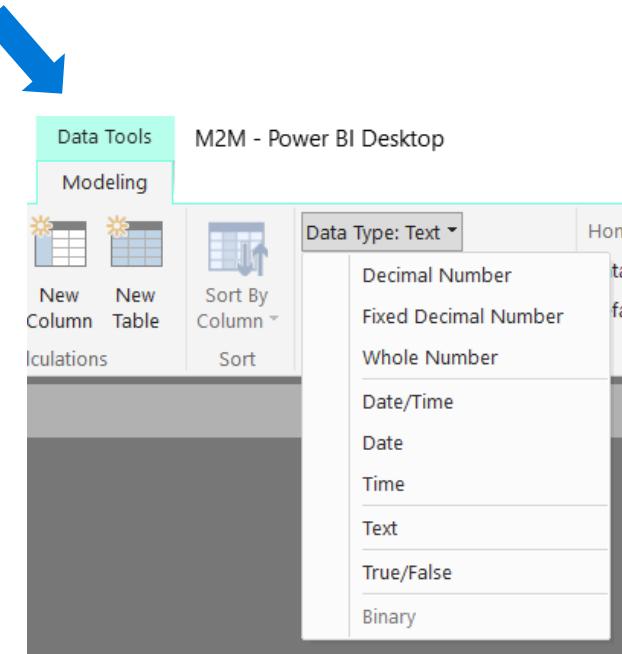
OK

Cancel

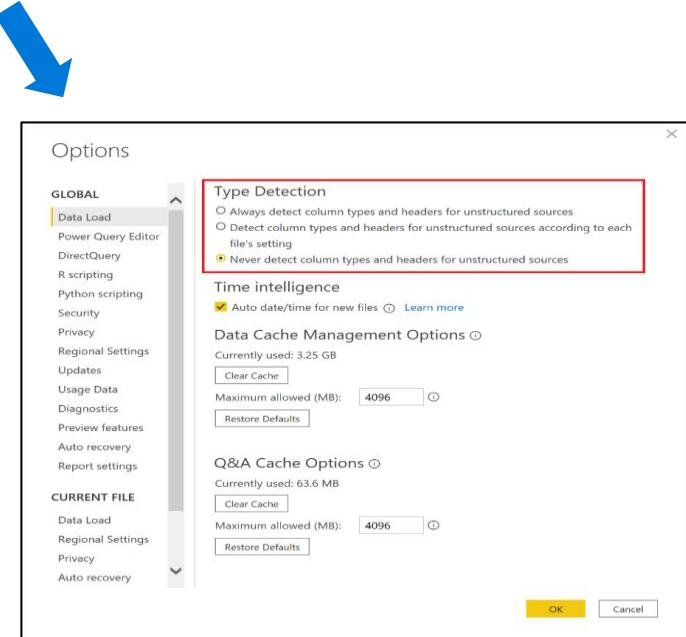
# Creating a Data Model

## Data Types

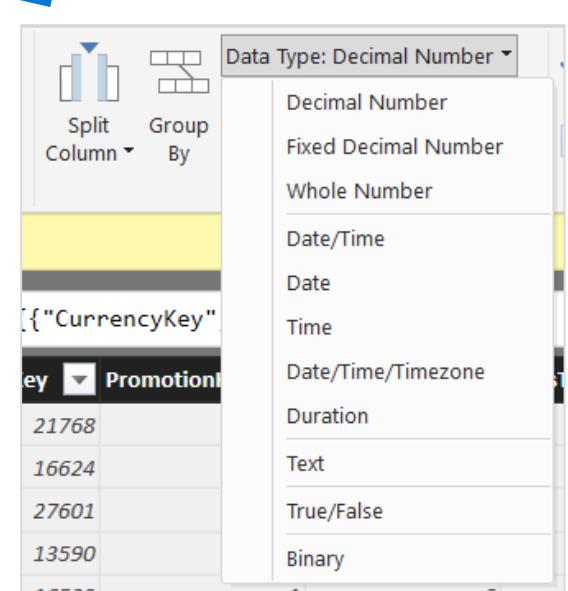
By default, Power BI Desktop **detects data types automatically** when loading the data for more efficient storage, calculations, and data visualization



Automatic type detection option can be **disabled globally** which can be found under **Options** dialog (under **Global > Data Load**)



Data types can be set both at the **Query Editor** and in the **Modeling** tab



Recommended to set data type in Query Editor

# Creating a Data Model

## Data Types

### Numeric Types

- **Decimal Number** – eight byte floating point number. The largest value is 15 digits long and decimal separator can occur anywhere.
- **Fixed Decimal Number** – It allows 19 digits with 4 digits of precision. The largest value is 922,337,203,685,477.5807 . Digits to the right of the 4 digits of precision are truncated. Good to avoid rounding errors.
- **Whole Number** – eight byte integer value. It has no decimal places. It allows 19 digits from -2 63 to 2 63-1 and allows the largest value of the numeric types.
- **Percentage** – An input value such as "5%" will be automatically recognized as a Percentage value and converted to a 2-digit precision decimal number (i.e. 0.05).

### Text type

### True/False type

### Date/Time Types

### Date/Time – Dates with time from year 1900 to 9999

- **Date** – A date without a time portion
- **Time** – Only the time portion
- **Date/Time/Timezone** – Represents a UTC Date/Time
- **Duration** – The difference between two Date/Time or Time fields

### Blank/Nulls type – represents and replaces SQL nulls.

# Creating a Data Model

## Data Categorization

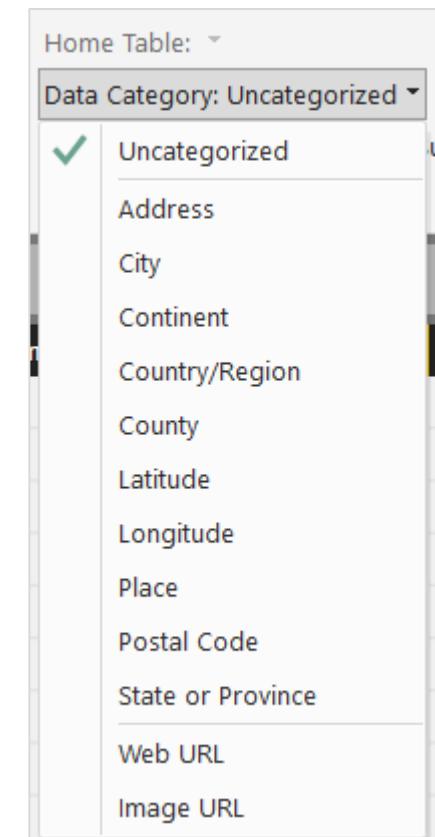
Categorizing the fields is **relevant** for **geographical** fields or for **image** or **URL** fields

In the modeling tab a field can be classified at a certain **geographical** granularity or as an **image that comes from an URL** or a text field which is actually an **URL**.

Power BI desktop will **render data based on this categorization**

GeoCode	Sales Amount
AL	\$ 10,175,870.00
AR	\$ 4,351,530.00
AZ	\$ 6,114,241.00
CA	\$ 6,688,589.00
KY	\$ 53,832,611.00

Am I seeing country sales or US states' sales?  
Is AL, Albania or Alabama? Or, is AR, Argentina or Arkansas?



# Creating a Data Model

## Standard Data Formatting

- Built-in formats based on data types.**

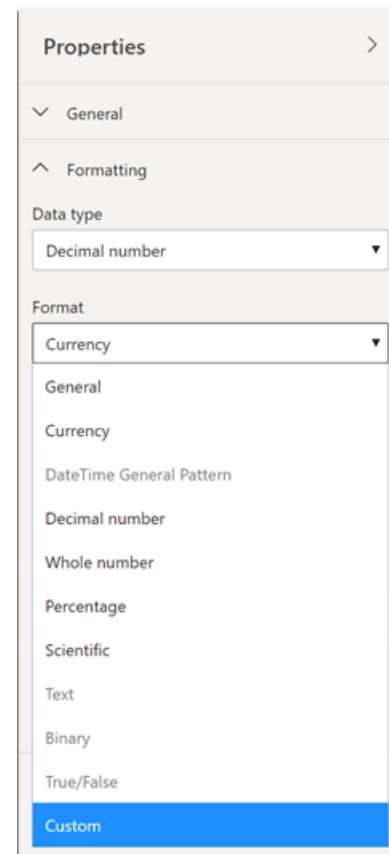
The screenshot shows the Power BI Desktop interface with the 'Column tools' ribbon tab selected. A dropdown menu for 'Format' is open, displaying the current date format as 'Wednesday, March 1'. The 'Date formats' section lists various date format options, such as 'Wednesday, March 14, 2001 (dddd, mmmm d, yyyy)' and 'March 14, 2001 (mmmm d, yyyy)'. To the right of the date formats, there is a list of year values and their corresponding year codes, such as '2013 CY2013' and '2013 CY2013'. The main workspace shows a table with columns for DateKey, Date, and YearKey.

DateKey	Date	YearKey	Year
20130701	Monday, July 1, 2013	2013	CY2013
20130702	Tuesday, July 2, 2013	2013	CY2013
20130703	Wednesday, July 3, 2013	2013	CY2013
20130704	Thursday, July 4, 2013	2013	CY2013
20130705	Friday, July 5, 2013	2013	CY2013
20130706	Saturday, July 6, 2013	2013	CY2013
20130707	Sunday, July 7, 2013	2013	CY2013
20130708	Monday, July 8, 2013	2013	CY2013
20130709	Tuesday, July 9, 2013	2013	CY2013
20130710	Wednesday, July 10, 2013	2013	CY2013
20130711	Thursday, July 11, 2013	2013	CY2013
20130712	Friday, July 12, 2013	2013	CY2013
20130713	Saturday, July 13, 2013	2013	CY2013
20130714	Sunday, July 14, 2013	2013	CY2013
20130715	Monday, July 15, 2013	2013	CY2013
20130716	Tuesday, July 16, 2013	2013	CY2013
20130717	Wednesday, July 17, 2013	2013	CY2013
20130718	Thursday, July 18, 2013	2013	CY2013
20130719	Friday, July 19, 2013	2013	CY2013
20130720	Saturday, July 20, 2013	2013-Jul-20	2013-Q3

# Creating a Data Model

## Custom Format Strings

- **Created in Model view.**
- **VBA style syntax.**



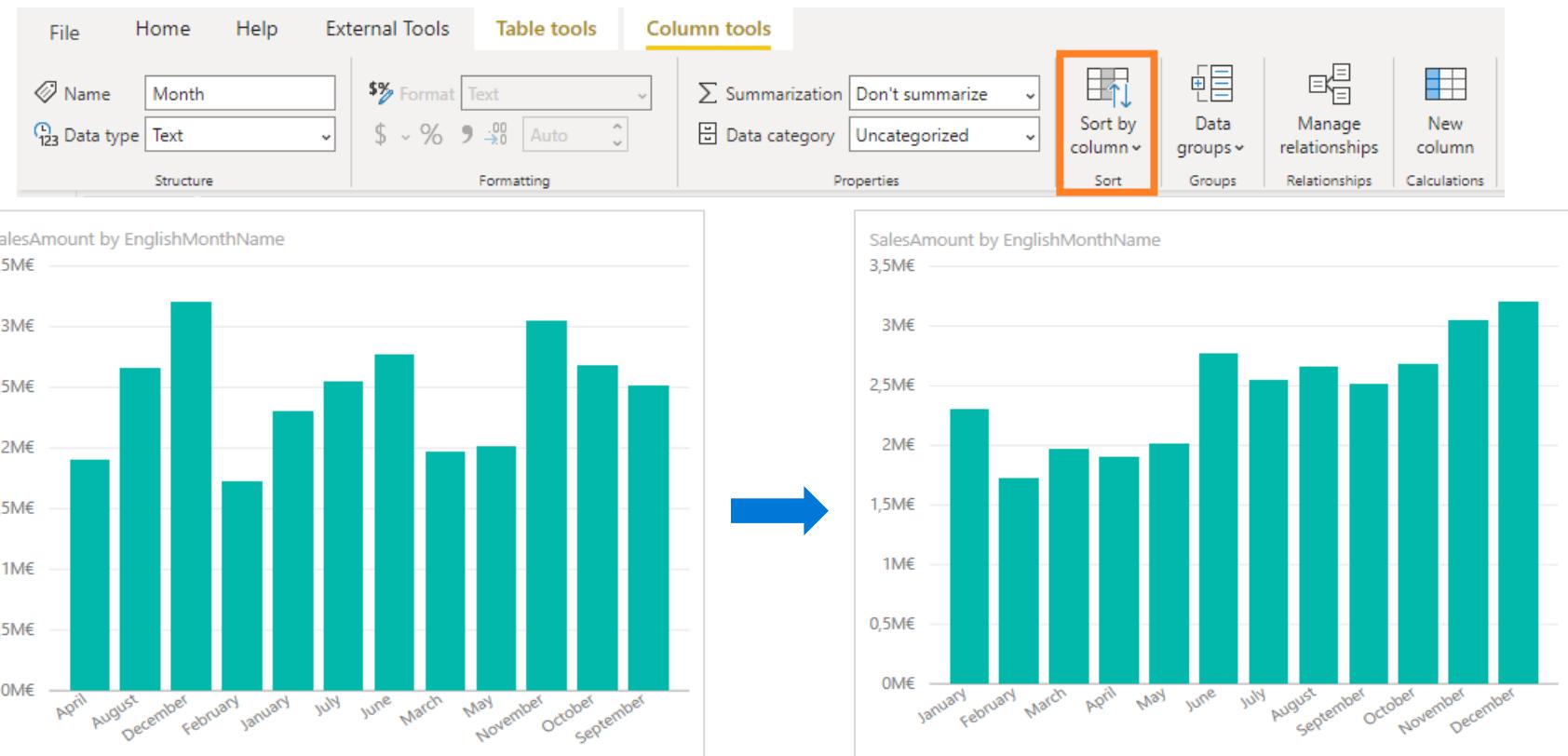
The screenshot shows two separate 'Custom format' dialog boxes. The top one displays the formula '#,##.##;(#,##.##);"Zero"; (123,456.8)' with its preview showing 'General Number (123,456.78)'. The bottom one displays the formula '\$#,##0;(\$#,##0)' with its preview showing 'Percent (34%)'. Both dialogs include a 'Learn more' link and a dropdown arrow.

# Creating a Data Model

## Sorting by Column

By **default**, Power BI desktop will sort data in a column based on the **alphabet**

It is possible to **use a separate column to determine sorting**



# Creating a Data Model

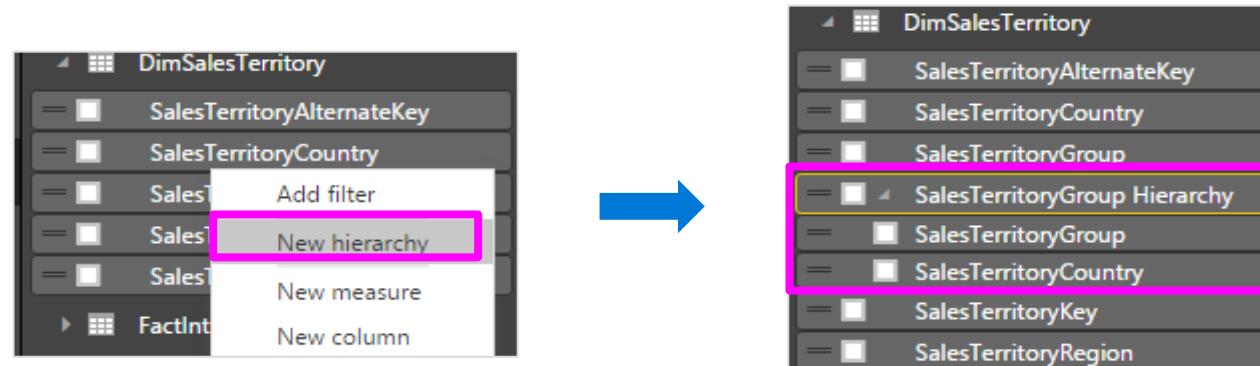
## Hierarchies

**Can be created in the Data View**

They must use attributes from a **single entity**

Can be **used in the different visualizations, allowing drill down**

Imported **Power Pivot models** or connected to **SSAS models with hierarchies** are also usable



# Creating a Data Model

Mark custom date table

---

You can **mark a custom table as a date table** (suppresses OOB date tables)

---

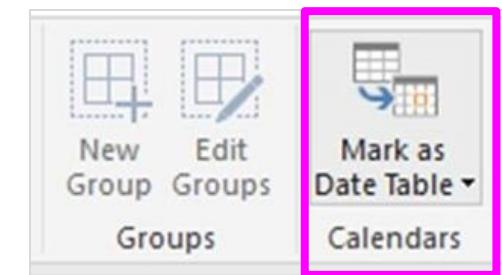
**Date hierarchies, time intelligence and Quick Measures** become enabled

---

Must contain at least **one column of type “Date” or “Date/Time”**

---

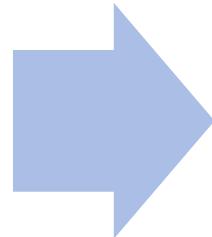
**Must be contiguous**



# Creating a Data Model

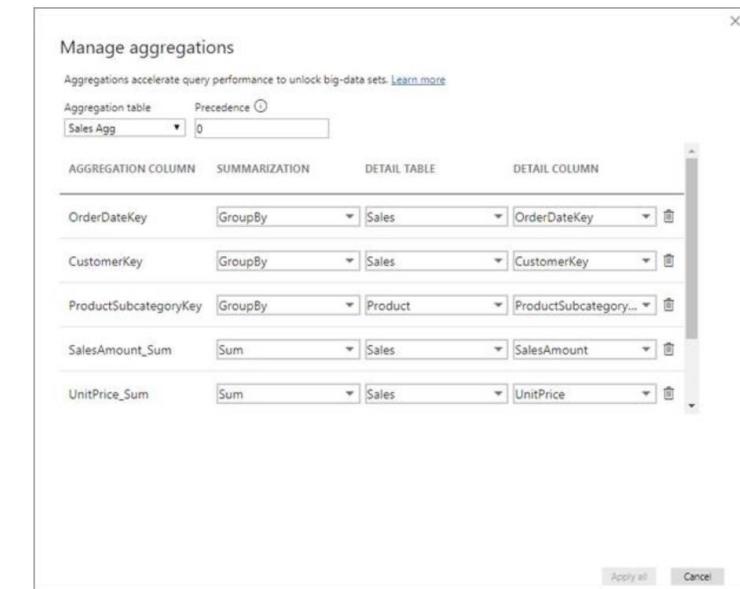
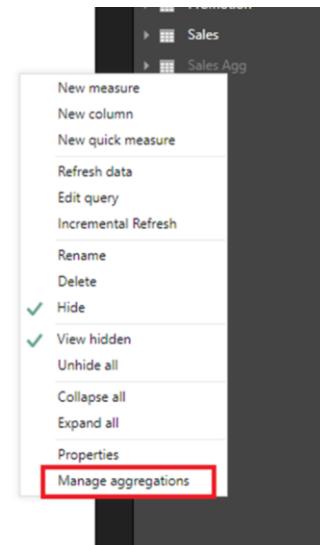
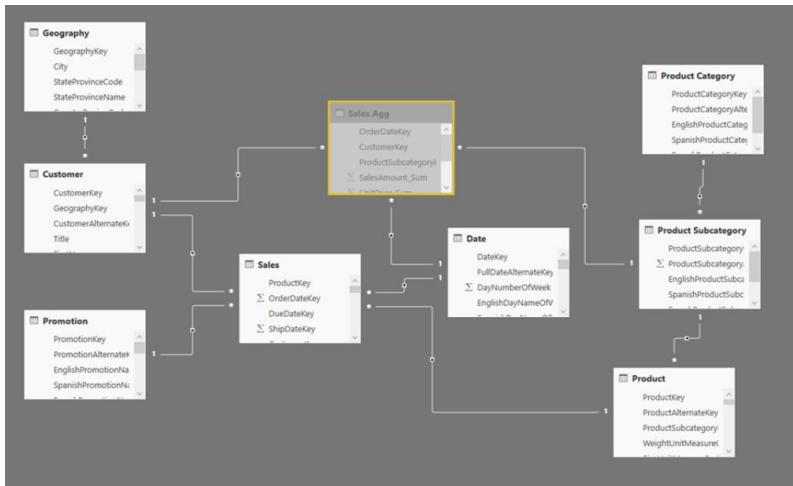
## Aggregations

**Aggregations** are a way to **store only summarized values in memory**, to provide the **fastest access** to **aggregated data without having to load** your entire dataset into Power BI



### Benefits of Aggregations:

- **Query performance** on large datasets
- Data **refresh optimization**
- Achieve **balanced architectures**



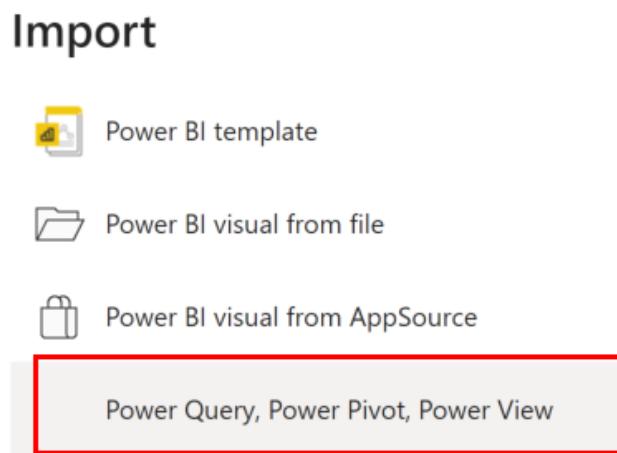
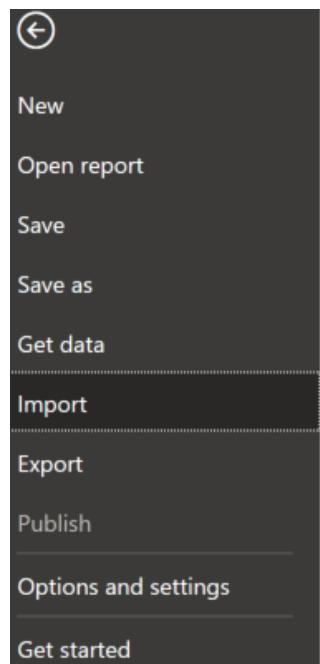
# Creating a Data Model

## Import From Microsoft Excel

If you already have **data models created in Excel**, you can import them into Power BI Desktop (opposite, not possible)

**Power Query, Power Pivot** and **Power View** content will be “migrated”

When **finished**, there is **no longer a dependency** on the Excel file



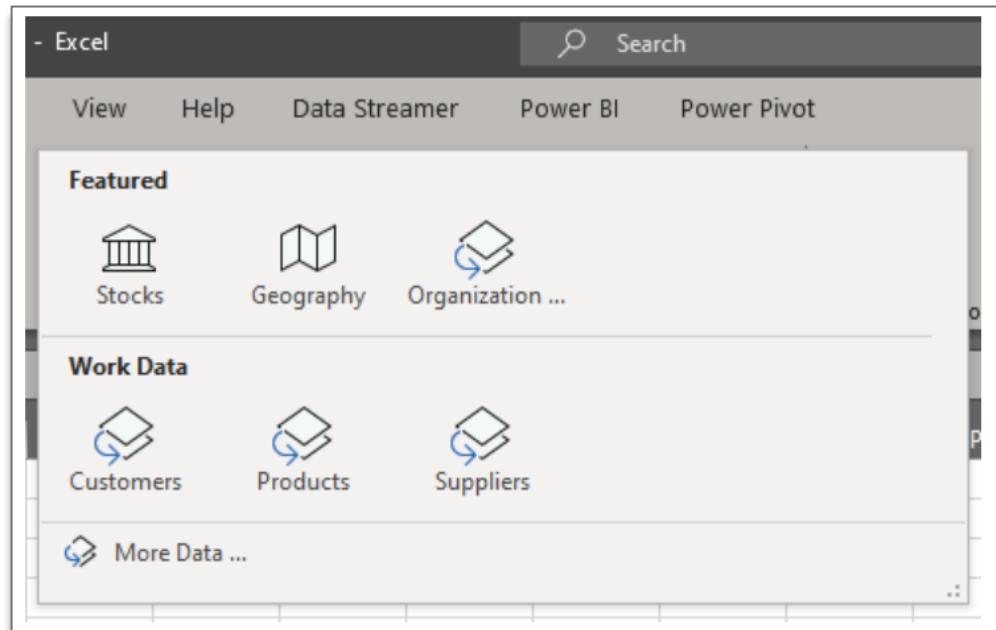
# Creating a Data Model

Curate Featured Tables for Excel ( Preview )

Curate **featured tables** to be referenced in Excel that will **enrich** your Excel sheets with authoritative enterprise data.

Enable members of your organization to **find** and **use relevant** and **refreshable data** to ultimately make better decisions

A featured table must have a **description**, a **row label**, and a **key column**.



Let's Stop N Shop	
Data retrieved 4/14/2020 12:35	
CustomerID	LETSS
ContactName	Jaime Yorres
ContactTitle	Owner
Address	87 Polk St. Suite 5
City	San Francisco
Region	CA
PostalCode	
Powered by Power BI	

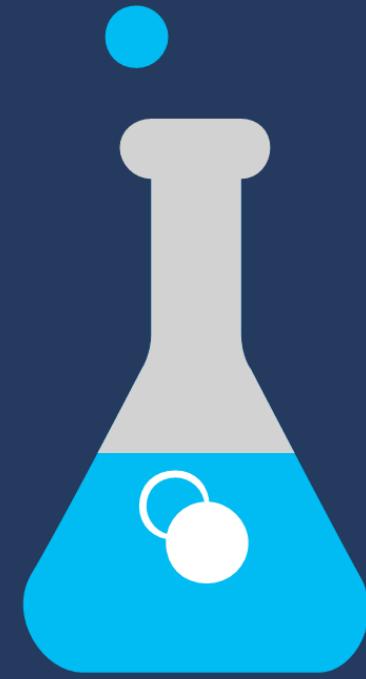
# Demonstration

## Data Model Quick Tour



# Power BI Desktop

- Creating the Data Model



LAB

**Questions?**



# Knowledge Check

What are the different relationship cardinality supported in Power BI data models?

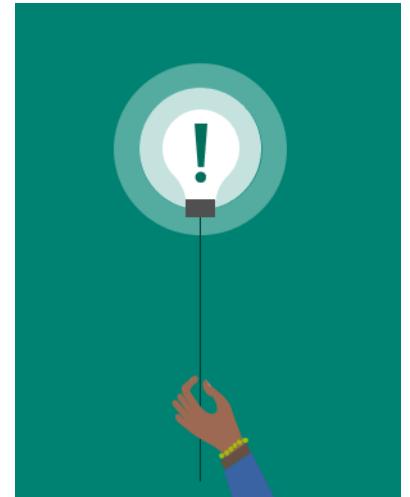
What is the impact of using “Both” filter direction?

# Lesson 5: Calculations

# Objectives

After completing this learning, you will be able to:

- Learn different calculation methods to prepare data



# Calculations

## DAX Basics

**Microsoft Excel like Language** that allows the extension of the model with additional business logic

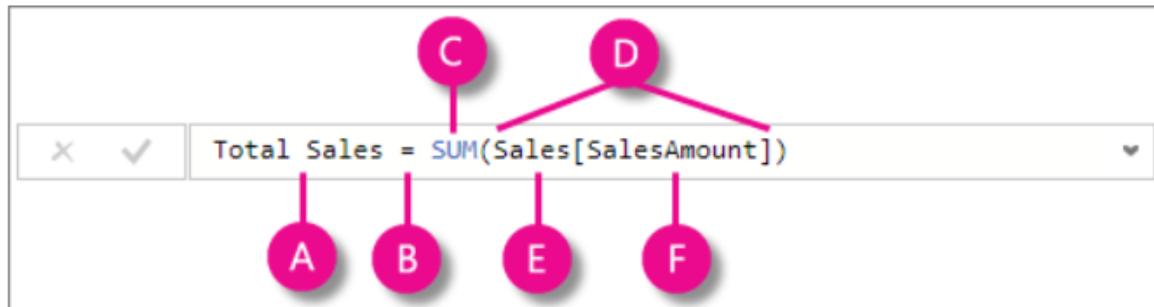
Ideal for **complex calculations** such as time intelligence, growth formulas, ratios, and more complex KPIs

Business rules can be materialized at the row level (Calculated Column) or can be calculated on-the-fly (Measures)

Certain capabilities might overlap with Formula Language

# Calculations

## DAX Basics – Syntax



- A An expression always **starts with the name of the calculation**
- B **The equal sign** indicates the **beginning of the formula**
- C **A function or a combination of functions** is applied which will **return a value**
- D **The arguments of the function** (can be a reference to columns or additional functions)
- E **The table** that is being referenced
- F **The column** that is being referenced for the specified table

# Calculations

## DAX Basics - Functions

A function always **references a column or a table**. Filters can be added to filter context of evaluation

A function always **returns a value or a table**. When a table is returned, further functions should be applied to obtain a value

Several **time intelligence** functions exist out of the box

Some Excel functions are also valid (like MONTH, FLOOR...)

Full reference of available functions here: <https://msdn.microsoft.com/en-us/library/ee634396.aspx>

```
Total Sales Amount Current = CALCULATE(SUM(FactInternetSales[SalesAmount]);FILTER(DimCustomer;DimCustomer[NumberCarsOwned]<3))
```

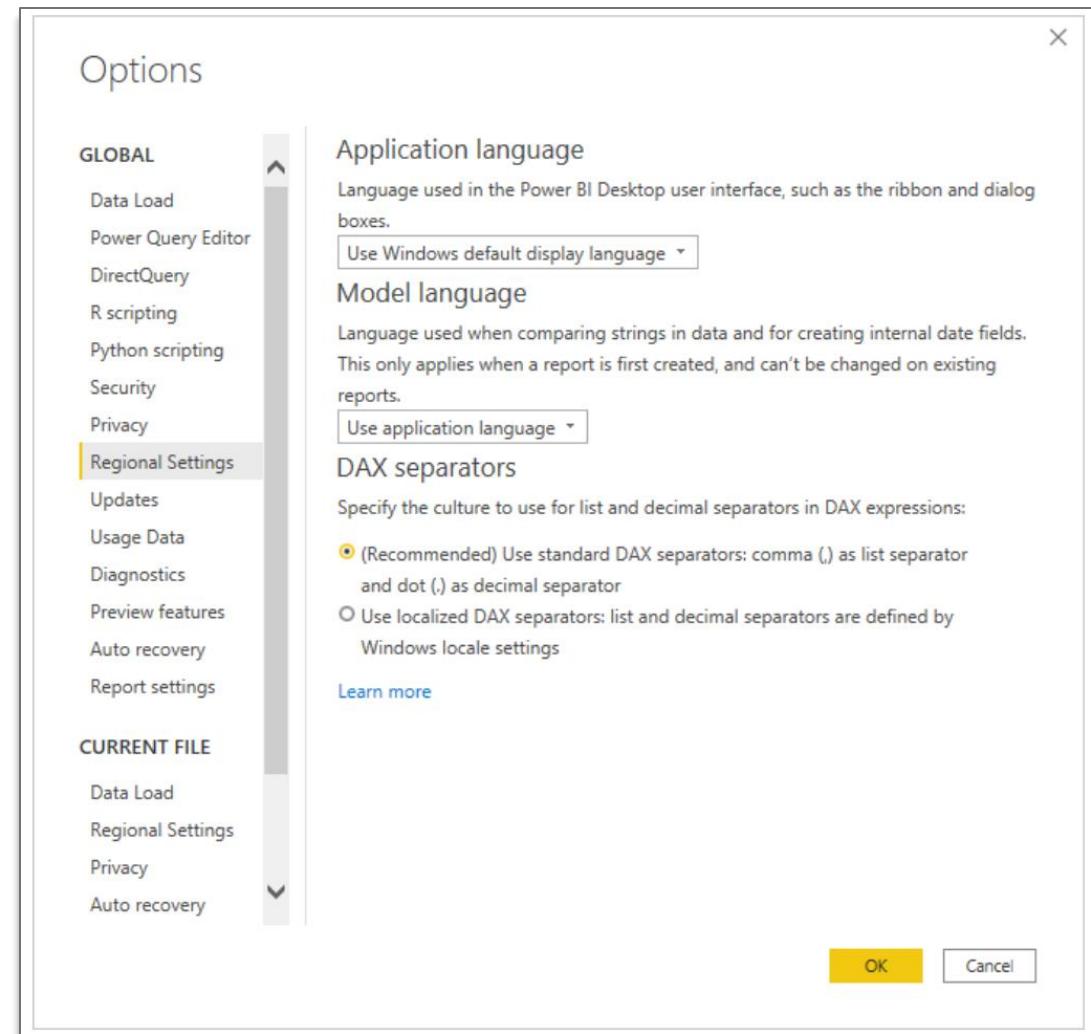
```
Total Sales Amount = CALCULATE([Total Sales Amount Current];DATESYTD(DimDate[FullDateAlternateKey]))
```

# Calculations

## DAX Basics – List Separator and Decimal Symbol

DAX authoring will default to using the **standard DAX** characters of **comma** as the **list separator** and **period** as the **decimal symbol**.

If you want to **default back** to the list separator and decimal symbols defined by your machine's Region settings, ,use DAX separators option under Options and settings > Options > Global > Regional settings:



# Calculations

## DAX Basics – Dax Editor

The editor has **keyboard shortcuts**, line numbers, and indent lines. The experience is very similar to what you have for other Microsoft editors, such as VS Code.

Alt+↑ / ↓	Move line up/down
Shift+Alt + ↓ / ↑	Copy line up/down
Ctrl+Enter	Insert line below
Ctrl+Shift+Enter	Insert line above
Ctrl+Shift+\ \\	Jump to matching bracket
Ctrl+] / [	Indent/outdent line
Alt+Click	Insert cursor
Ctrl+I	Select current line
Ctrl+Shift+L	Select all occurrences of current selection
Ctrl+F2	Select all occurrences of current word

# Calculations

## DAX Basics – Creating Calculations

Select the **Data View** and then the **Modeling tab** for the target table (or with the table contextual menu)

Type the **expression** in the **Formula Bar**

The screenshot shows the Power BI Data View interface. The top navigation bar includes File, Home, Help, External Tools, Table tools (which is the active tab), and Column tools. Below the navigation bar, there's a table named "Date". A blue arrow points from the "Table tools" tab in the ribbon to the "New Column" option in the context menu that has been opened over the "Month" column header. The context menu also lists New Measure, Refresh Data, Copy Table, Rename, Delete, Hide in Report View, and Unhide All. At the bottom of the screen, the formula bar displays the DAX expression: "Total Sales Amount = SUM(FactInternetSales[SalesAmount])".

DateKey	Date	Day	MonthKey	Month	QuarterKey	Quarter	YearKey	Year
20130701	Monday, July 1, 2013	2013-Jul-01	201307	2013-Jul	20133	CY2013-Q3	2013	CY2013
20130702	Tuesday, July 2, 2013	2013-Jul-02	201307	2013-Jul	20133	CY2013-Q3	2013	CY2013
20130703	Wednesday, July 3, 2013	2013-Jul-03	201307	2013-Jul	20133	CY2013-Q3	2013	CY2013
20130704	Thursday, July 4, 2013	2013-Jul-04	201307	2013-Jul	20133	CY2013-Q3	2013	CY2013
20130705	Friday, July 5, 2013	2013-Jul-05	201307	2013-Jul	20133	CY2013-Q3	2013	CY2013
20130706	Saturday, July 6, 2013	2013-Jul-06	201307	2013-Jul	20133	CY2013-Q3	2013	CY2013
20130707	Sunday, July 7, 2013	2013-Jul-07	201307	2013-Jul	20133	CY2013-Q3	2013	CY2013
20130708	Monday, July 8, 2013	2013-Jul-08	201307	2013-Jul	20133	CY2013-Q3	2013	CY2013
20130709	Tuesday, July 9, 2013	2013-Jul-09	201307	2013-Jul	20133	CY2013-Q3	2013	CY2013
20130710	Wednesday, July 10, 2013	2013-Jul-10	201307	2013-Jul	20133	CY2013-Q3	2013	CY2013
20130711	Thursday, July 11, 2013	2013-Jul-11	201307	2013-Jul	20133	CY2013-Q3	2013	CY2013
20130712	Friday, July 12, 2013	2013-Jul-12	201307	2013-Jul	20133	CY2013-Q3	2013	CY2013

Total Sales Amount = **SUM(FactInternetSales[SalesAmount])**

# Calculations

DAX Basics – How are calculations applied?

**Row context** can be thought of as the current row - like an iterator

**Filter context** determines the conditions that filter data, for a particular visualization, and can be applied on top of Row Context

Sales Minus Tax = FactInternetSales[SalesAmount]-FactInternetSales[TaxAmt]

Row Context

CalendarYear	Gender	Sales Minus Tax
2013	F	7.583.041 €
2013	M	7.460.384 €
2011	F	3.278.038 €
2011	M	3.231.445 €
2012	F	2.726.550 €
2012	M	2.648.535 €
2014	F	21.638 €
2010	M	20.687 €
2014	M	20.400 €
2010	F	19.259 €
Total		27.009.982 €

This cell corresponds to the sum of "Sales Minus Tax" for all rows that belong to the Calendar Year "2012" and Gender is "M"

# Calculations

## Calculated Columns

They are **persisted** in the model and are calculated row-by-row (Row Context)

They **increase the size and memory requirements** for the model

They can be **re-used in other calculations**

**DAX Functions** can be used in their definition (beware of aggregations)

Sales Minus Tax = FactInternetSales[SalesAmount]-FactInternetSales[TaxAmt]

	SalesAmount	TaxAmt	Freight	Sales Minus Tax
1	4,99 €	0,3992 €	0,1248 €	4,5908 €
2	4,99 €	0,3992 €	0,1248 €	4,5908 €
3	4,99 €	0,3992 €	0,1248 €	4,5908 €
4	4,99 €	0,3992 €	0,1248 €	4,5908 €
5	4,99 €	0,3992 €	0,1248 €	4,5908 €
6	4,99 €	0,3992 €	0,1248 €	4,5908 €

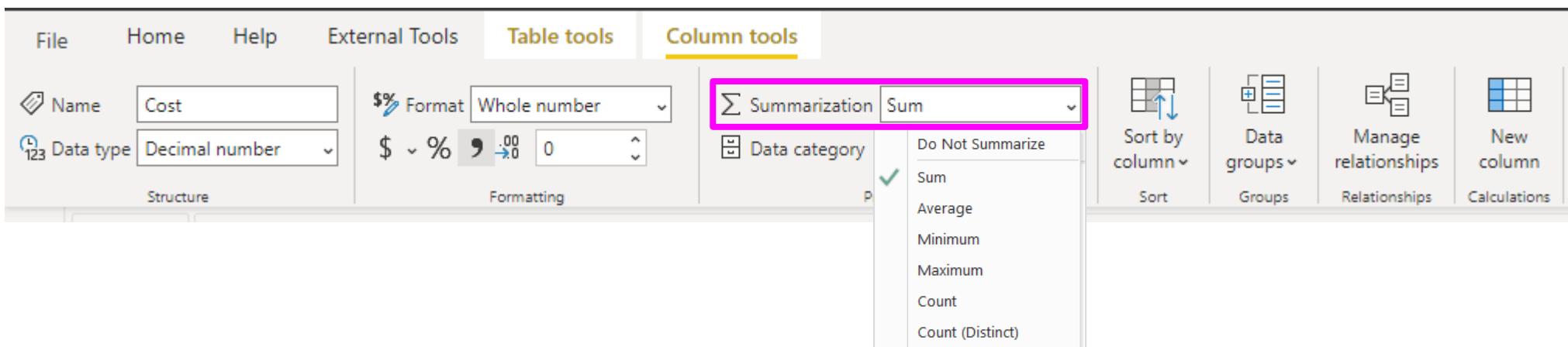
# Calculations

## Calculated Columns

A **calculation** or a **numeric field** has a “**Default Summarization**” – defines how it is aggregated when placed on a visualization, by default

A Calculated Column with a “Default Summarization” of “Do not summarize” will appear with  ScenarioKey

With a **different summarization** will appear with  Sales Minus Tax



# Calculations

## Measures

Measures are **calculated when they are used** in a particular visualization

And they **can also be calculated on a row-by-row basis**

They can be **implicit** – an aggregation of a field with “Default Summarization” different from “Do Not Summarize”

 SalesAmount

They can be **explicit**, where they are the result of a DAX expression. You cannot control the aggregation for this type.

 Total Sales Amount

# Calculations

## Measures

They can (and should) be **referenced from other measures** – this is a best practice

Measures are **evaluated for each cell** they appear in (Filter Context)

The screenshot shows a Power BI report interface. At the top left is a dropdown menu labeled "Gender" with options "F" and "M". Below it is a table with the following data:

CalendarYear	EnglishPromotionCategory	Average Sales per Transaction
2011	Reseller	2,887 €
2012	Reseller	1,857 €
2011	No Discount	1,827 €
2013	Reseller	1,605 €
2012	No Discount	1,240 €
2013	No Discount	1,166 €
Total		1,326 €

CalendarYear=2012  
Promotion="Reseller"  
Gender="M"

The total is also calculated independently. It is not an aggregation of the rows.

# Calculations

## Measures

**Data categorization** can be set on measures. Can do things like:

- use measures to **dynamically create URLs** and mark the data category as a Web URL; especially useful to create link off to other Power BI reports with URL filter parameters.
- create DAX measures to create interesting insights such as **dynamic sparklines using SVG images**

Home Table: Sales ▾

Data Category: Uncategorized ▾

	Sales	Target Sales	Sparkline Column	Sparkline Line
Uncategorized				
Address				
City				
Continent				
Country/Region				
County				
Latitude				
Longitude				
Place				
Postal Code				
State or Province				
Web URL				
Image URL				
Barcode				



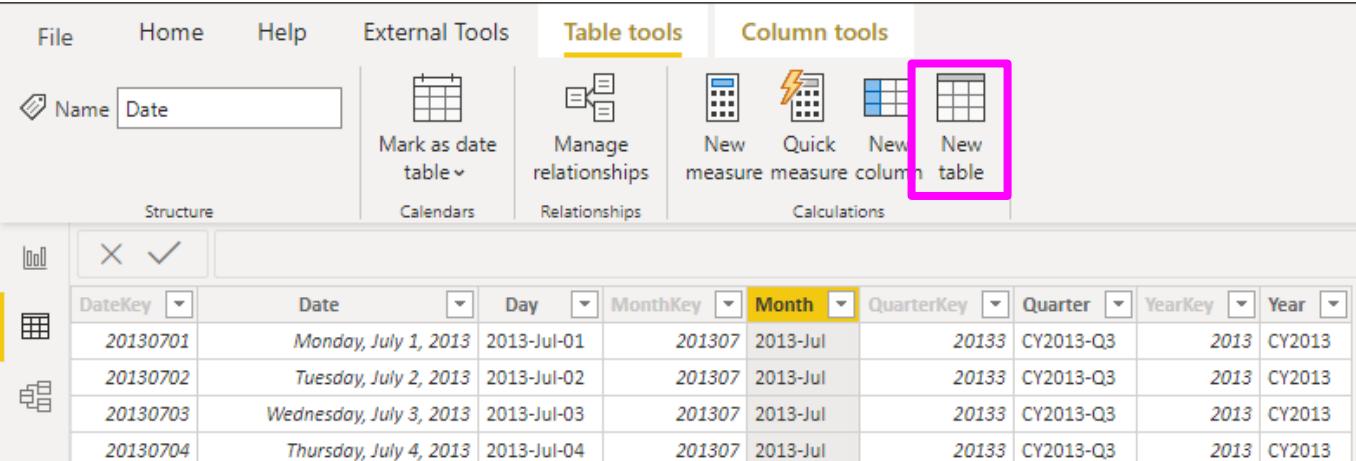
# Calculations

## Calculated Tables

Become new tables which are the **result of a DAX Expression**

They behave **like regular tables** and **created via Modeling Tab**

New tables **can also be created in the Query Editor** with Formula Language **or by typing or pasting content**



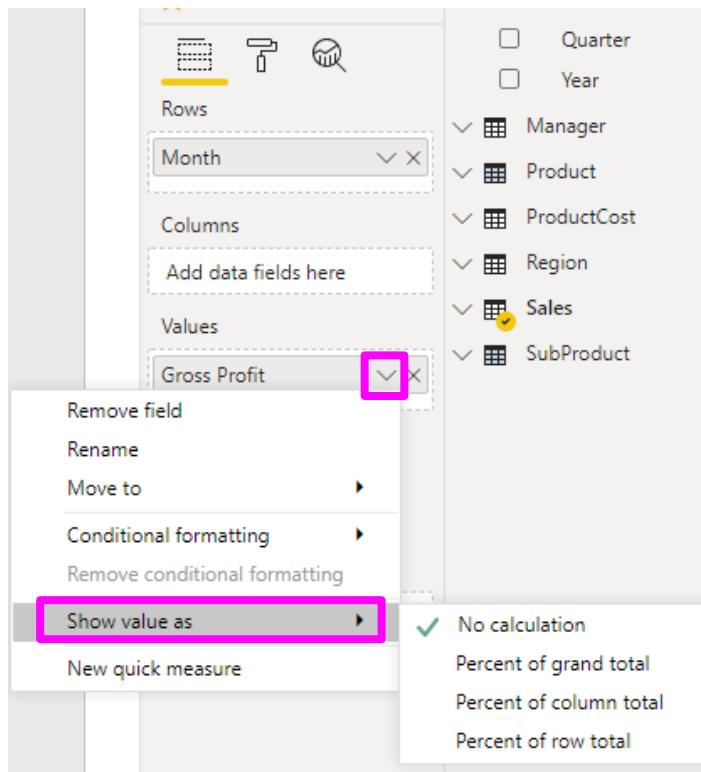
DateKey	Date	Day	MonthKey	Month	QuarterKey	Quarter	YearKey	Year
20130701	Monday, July 1, 2013	2013-Jul-01	201307	2013-Jul	20133	CY2013-Q3	2013	CY2013
20130702	Tuesday, July 2, 2013	2013-Jul-02	201307	2013-Jul	20133	CY2013-Q3	2013	CY2013
20130703	Wednesday, July 3, 2013	2013-Jul-03	201307	2013-Jul	20133	CY2013-Q3	2013	CY2013
20130704	Thursday, July 4, 2013	2013-Jul-04	201307	2013-Jul	20133	CY2013-Q3	2013	CY2013

# Calculations

Show value as

**Out-of-the-box calculations** that can be applied without coding

Accessible **via field well**, you can change how a numerical value or measure is displayed



# Calculations

## Advanced Calculations

- **CALCULATE(<measure expression>, <filter1>, <filter2>, ...)** allows us to change the filter context

No Discount Sales =

```
CALCULATE (
    SUM ( FactResellerSales[SalesAmount] );
    DimPromotion[EnglishPromotionName] = "No Discount"
)
```

CalendarYear	EnglishPromotionCategory	No Discount Sales	SalesAmount
2011	No Discount	15,772,060 €	15,772,060 €
2011	Reseller		516,381 €
2012	No Discount	26,609,942 €	26,609,942 €
2012	Reseller		1,311,727 €
2013	No Discount	32,645,904 €	32,645,904 €
2013	Reseller		3,594,579 €
<b>Total</b>		<b>75,027,907 €</b>	<b>80,450,596 €</b>

# Calculations

## Advanced Calculations

**FILTER** allows more complex filtering but is less performant than **CALCULATE**

Should be **used to filter on smaller tables** like Dimensions

It **follows the relationships** in order to filter, so its results are more user-friendly

CalendarYear	EnglishPromotionCategory	DiscountPct	Sales with Big Discount	Sales with Big Discount FILTER
2013	Reseller	0.02	1,060,673 €	
2013	Reseller	0.05	1,060,673 €	
2013	Reseller	0.10	1,060,673 €	
2013	Reseller	0.15	1,060,673 €	453,442 €
2013	Reseller	0.20	1,060,673 €	581,331 €
2013	Reseller	0.30	1,060,673 €	
2013	Reseller	0.35	1,060,673 €	
2013	Reseller	0.40	1,060,673 €	25,899 €
<b>Total</b>			1,060,673 €	1,060,673 €

# Calculations

## Advanced Calculations

Sales with Big Discount =

```
CALCULATE (
    SUM ( FactResellerSales[SalesAmount] );
    DimPromotion[DiscountPct] > 0.1
)
```

Sales with Big Discount FILTER =

```
CALCULATE (
    SUM ( FactResellerSales[SalesAmount] );
    FILTER ( DimPromotion; DimPromotion[DiscountPct] > 0.1 )
)
```

# Calculations

## Advanced Calculations

- **ALL removes the filter applied to a table or column.**
- Useful for **ratio-to-parent calculations**

Percentage of Product Net Revenue =

```
DIVIDE(  
    SUM(Sales[Net Revenue]);  
    CALCULATE(  
        SUM(Sales[Net Revenue]);  
        ALL(Product[Product Name])  
    )  
)
```

Product Item Group	Product Name	Net Revenue	Percentage of Product Net Revenue
Helicopter	Tailspin Heli - Max Pro Flight - 6ch	36,253,883.30	71.72 %
	6CCP-A Helicopter	8,662,982.90	17.14 %
	Tailspin Heli - Co-Ax Pro Mk I - 4ch	2,908,014.50	5.75 %
	4CAX-B Helicopter	1,336,857.20	2.64 %
	3CAX-B Helicopter	1,034,853.80	2.05 %
	3CFP-I Helicopter	198,883.20	0.39 %
	Tailspin Heli - Pro Mk III - 5ch	78,671.70	0.16 %
	4CFP-I Helicopter	77,308.00	0.15 %
	Total	50,551,454.60	100.00 %

# Calculations

## Advanced Calculations

---

### **X Functions** (SUMX, MAXX, MINX, AVERAGEX, COUNTX, COUNTAX)

---

Parameters (<table or table expression>,<arithmetic expression>)

---

It will **iterate each row** on the first table and apply the arithmetic expression.

---

Useful for **row-by-row calculations, correcting TOTALS and hidden calculations**

---

Might, potentially, **be slower than SUM**

# Calculations

## Advanced Calculations

### X Functions - Row-by-row

SalesAmountwithSUMX =

SUMX (

FactResellerSales;

FactResellerSales[OrderQuantity] \* FactResellerSales[UnitPrice]

)

SalesOrderNumber	SalesOrderLineNumber	RevisionNumber	OrderQuantity	UnitPrice
SO43912	1	1	1	874,794 €
SO43912	4	1	1	419,4589 €
SO43912	8	1	1	874,794 €
SO43912	10	1	1	183,9382 €
SO43912	13	1	1	2 146,962 €
SO43912	14	1	1	20,1865 €

# Calculations

## Advanced Calculations

- **RANKX**(<table expression>;<arithmetic expression>;<sort order>;<tie handler>)
- Gets an ordinal position for the selected column

ProductRank =

`RANKX ( ALL ( DimProduct[ProductAlternateKey] ); [Sales];; DESC; SKIP )`

ProductAlternateKey	ProductRank	Sales
BK-M68B-38	1	3.105.726,66 €
BK-M68B-42	2	2.646.352,67 €
BK-M68B-46	6	1.936.203,67 €
BK-M68S-38	3	2.354.215,23 €
BK-M68S-42	4	2.181.044,29 €
BK-M68S-46	5	2.133.156,84 €
BK-R79Y-48	9	1.380.253,88 €
BK-R89B-44	7	1.888.480,05 €
BK-R89B-48	8	1.656.449,69 €
BK-T79U-60	10	1.370.784,22 €
<b>Total</b>	<b>1</b>	<b>20.652.667,21 €</b>

# Calculations

## Advanced Calculations

- **HASONEVALUE** lets us test if only a single element is selected for a column

ProductRankEnhanced =

```
IF (
    HASONEVALUE ( DimProduct[ProductAlternateKey] );
    RANKX ( ALL ( DimProduct[ProductAlternateKey] ); [Sales];; DESC; SKIP );
    BLANK ()
)
```

ProductAlternateKey	ProductRankEnhanced	Sales
BK-M68B-38	1	3.105.726,66 €
BK-M68B-42	2	2.646.352,67 €
BK-M68S-38	3	2.354.215,23 €
BK-M68S-42	4	2.181.044,29 €
BK-M68S-46	5	2.133.156,84 €
BK-M68B-46	6	1.936.203,67 €
BK-R89B-44	7	1.888.480,05 €
BK-R89B-48	8	1.656.449,69 €
BK-R79Y-48	9	1.380.253,88 €
BK-T79U-60	10	1.370.784,22 €
Total		20.652.667,21 €

# Calculations

## Advanced Calculations

- **TOPN(<n>;<table>;<order by expression>)** returns the top N rows for a table

`TOPN ( 10; VALUES ( DimProduct[ProductAlternateKey] ); [Sales]; DESC )`

- The **VALUES** function is required to remove duplicates, if they exist. Not needed if they don't

ProductAlternateKey
BK-M68B-38
BK-M68B-42
BK-M68B-46
BK-M68S-38
BK-M68S-42
BK-M68S-46
BK-R79Y-48
BK-R89B-44
BK-R89B-48
BK-T79U-60

ProductAlternateKey	ProductRankEnhanced	Sales
BK-M68B-38	1	3.105.726,66 €
BK-M68B-42	2	2.646.352,67 €
BK-M68S-38	3	2.354.215,23 €
BK-M68S-42	4	2.181.044,29 €
BK-M68S-46	5	2.133.156,84 €
BK-M68B-46	6	1.936.203,67 €
BK-R89B-44	7	1.888.480,05 €
BK-R89B-48	8	1.656.449,69 €
BK-R79Y-48	9	1.380.253,88 €
BK-T79U-60	10	1.370.784,22 €
Total		20.652.667,21 €

# Calculations

## Advanced Calculations

- **TOPN** does not return a scalar
- It should be then **used as context to return a scalar**

TOP10Products =

```
CALCULATE (
    [Sales];
    TOPN ( 10; VALUES ( DimProduct[ProductAlternateKey] );
    [Sales]; DESC )
```

EnglishMonthName	TOP10Products	Sales	WeightTop10ProductsInSales
January	2.427.189,37 €	9.352.570,54 €	25,95%
February	2.225.028,54 €	6.932.933,24 €	32,09%
March	1.543.855,07 €	6.094.888,16 €	25,33%
April	2.133.344,04 €	6.536.977,73 €	32,64%
May	2.726.482,77 €	9.723.242,29 €	28,04%
June	1.077.145,23 €	2.980.089,16 €	36,14%
July	1.822.420,83 €	5.797.264,08 €	31,44%
August	2.235.315,93 €	7.658.678,04 €	29,19%
September	1.370.352,50 €	4.954.903,60 €	27,66%
October	2.004.131,14 €	8.464.470,18 €	23,68%
November	1.743.329,10 €	6.405.911,33 €	27,21%
December	1.531.166,72 €	5.548.668,64 €	27,60%
Total	20.652.667,21 €	80.450.596,98 €	25,67%

# Calculations

## Advanced Calculations – Support for Excel Financial Functions

Almost all **financial functions** that you know and love from **Excel** are now available in **DAX**.

The functions have the **same** name and signature as their **Excel** counterpart.

These financial functions make it easier to build **financial reports** and perform otherwise complex calculations in Power BI

# Calculations

## Advanced Calculations – Time Intelligence

Many DAX functions exist to support time calculations. **There are two types:**

- Functions that **require a CALCULATE**

```
SalesYTD =  
CALCULATE ( [Sales]; DATESYTD ( DimDate[FullDateAlternateKey] ) )
```

- Functions that **return a scalar** (syntactic sugar)

```
SalesYTDNoCalculate =  
TOTALYTD ( [Sales]; DimDate[FullDateAlternateKey] )
```

# Calculations

## Advanced Calculations – Time Intelligence

```
SalesLastMonth =  
CALCULATE ( [Sales];  
DATEADD ( DimDate[FullDateAlternateKey]; -1; MONTH ) )
```

```
SalesSamePeriodLastYear =  
CALCULATE ( [Sales];  
SAMEPERIODLASTYEAR ( DimDate[FullDateAlternateKey] ) )
```

```
MonthOverMonth =  
DIVIDE ( ( [Sales] - [SalesLastMonth] ); [SalesLastMonth] )
```

```
SalesSinceEver =  
CALCULATE ( [Sales];  
DATESBETWEEN ( DimDate[FullDateAlternateKey];  
FIRSTDATE ( ALL ( DimDate[FullDateAlternateKey] ) );  
LASTDATE ( DimDate[FullDateAlternateKey] ) )  
)
```

# Calculations

## Advanced Calculations – Time Intelligence

To enable time intelligence with your date tables, two conditions are necessary:

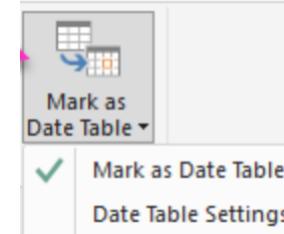
The **relationship** between the fact table and the date table must be done through a **date field on both sides**; Otherwise, the date table should be exclusively **marked as Date Table**.

The **calculation must then target the date column** on the date dimension which must be **contiguous**

FactResellerSales							
TaxAmt	Freight	CarrierTrackingNumber	CustomerPONumber	OrderDate	DueDate	ShipDate	
69,9835 €	21,8699 €	69B6-46F8-A4	POT656175368	29 de janeiro de 2011	10/02/2011	05/02/2011	
33,5567 €	10,4865 €	69B6-46F8-A4	POT656175368	29 de janeiro de 2011	10/02/2011	05/02/2011	
69,9835 €	21,8699 €	69B6-46F8-A4	POT656175368	29 de janeiro de 2011	10/02/2011	05/02/2011	

DimDate							
DateKey	FullDateAlternateKey	DayNumberOfWeek	EnglishDayNameOfWeek	SpanishDayNameOfWeek	FrenchDayNameOfWeek	GermanDayNameOfWeek	PortugueseDayNameOfWeek
20100701	1 de julho de 2010	5	Thursday	Jueves	Jeudi	Donnerstag	Sexta-feira
20100702	2 de julho de 2010	6	Friday	Vierernes	Vendredi	Freitag	Sábado
20100703	3 de julho de 2010	7	Saturday	Sábadoo	Samedi	Samstag	Domingo



CalendarYear	MonthNumberOfYear	SalesAmount	SalesYTD	SalesYTDNoCalculate
2013	1	4,162,825 €	4,162,825 €	4,162,825 €
	2	3,974,516 €	8,137,341 €	8,137,341 €
	3	2,260,306 €	10,397,647 €	10,397,647 €
	4	3,452,972 €	13,850,620 €	13,850,620 €
	5	3,465,744 €	17,316,365 €	17,316,365 €
	6	1,649,974 €	18,966,339 €	18,966,339 €
	7	2,681,169 €	21,647,509 €	21,647,509 €
	8	2,709,540 €	24,357,050 €	24,357,050 €
	9	2,189,338 €	26,546,388 €	26,546,388 €
	10	3,284,497 €	29,830,886 €	29,830,886 €
	11	3,372,403 €	33,203,289 €	33,203,289 €
	12		33,203,289 €	33,203,289 €
	Total		33,203,289 €	33,203,289 €
			33,203,289 €	33,203,289 €

# Calculations

## Advanced Calculations – Time Intelligence with Auto Date/Time

- **A hierarchy is automatically generated** for each date field on each table
- Calculations need to be done with the “in-line” notation

```
SalesAmountYTD =  
CALCULATE (  
    SUM ( FactInternetSales[SalesAmount] );  
    DATESYTD ( FactInternetSales[OrderDate].[Date] )  
)
```

# Calculations

## Advanced Calculations – Note on Auto date/time

Auto date/time is not **recommended for large tables** with many different date attributes and/or broad range of dates.

This feature **can bloat the data model** by creating separate internal date dimension tables for each date attribute.

Can be disabled under **Data Load** settings.

For large tables, **bring your own Date table** and mark it as Date Table.

# Calculations

## DAX Variables

Variables makes **writing DAX code easier, increases readability, reusability and maintenance** of DAX code.

Variables can **improve query performance**

Syntax:

- **VAR** keyword introduces the definition of a variable
- **RETURN** keyword defines the expression to return as the result

*myFormula =*

**VAR** = *VariableName1 = <some valid DAX formula>*

**VAR** = *VariableName2 = <some other valid DAX formula>*

**RETURN** *<another valid DAX formula that can use VariableName1  
and VariableName2 as part of the expression>*

# Calculations

## DAX Variables

---

Can have **as many variables** in an expression as required, and each one has its own definition

---

Variables can contain **scalar values** or **tables**

---

DAX **evaluates variables in the context of their definition**, not in the one where they are used.

---

Once the variable has been assigned a value, that value cannot change during the **execution of the RETURN** portion of the formula

---

The **results** of the variable calculations **are cached** for reuse inside the formula

---

Variables **can refer other variables, measures or calculated columns**

---

`VAR myScalarValue = SUM(Sales[Extended Amount])`

`VAR myTable = FILTER(Customer, Customer[Post Code] = 50210)`

# Calculations

## DAX Variables (example)

To calculate **a percentage of year-over-year growth** without using a variable, we could create three separate measures:

**Sum of SalesAmount** = `SUM(SalesTable[SalesAmount])`

**SalesAmount PreviousYear** = `CALCULATE([Sum of SalesAmount], SAMEPERIODLASTYEAR(Calendar[Date]))`

**Sum of SalesAmount YoY%** = `DIVIDE(([Sum of SalesAmount] – [SalesAmount PreviousYear]), [Sum of SalesAmount])`

Using a variable, we can create a **single measure** that calculates the same result

`YOY% = VAR CurrentSales = SUM(Sales[Net Revenue])`

`VAR SalesLastYear = CALCULATE(SUM(Sales[Net Revenue]), SAMEPERIODLASTYEAR('Date'[Date]))`

`RETURN IF(CurrentSales, DIVIDE(CurrentSales - SalesLastYear, CurrentSales))`

Advantages:

- More **readable** code
- Better **query performance** (result of expression is stored in the variable upon declaration and doesn't have to be recalculated each time it is used)

# Calculations

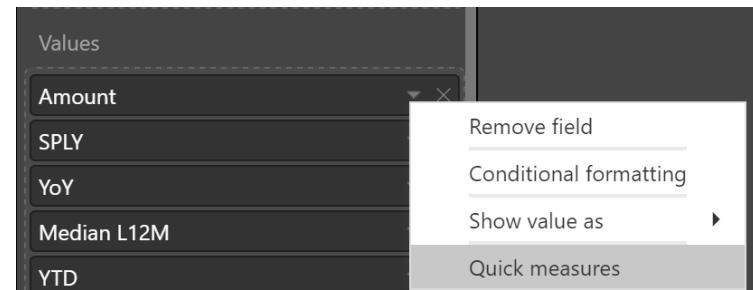
## Quick Measures

Allow you to **build calculations, without knowing DAX**

Also **works with** live connection mode against **SSAS models**.

Available calculations are organized in **6 categories**:

- Aggregate within category
- Filters and baselines
- Time Intelligence
- Running Total
- Mathematical Operations
- Text



Start with an element on the field well or on the field list and create calculation

It's great for **learning!**

# Demonstration

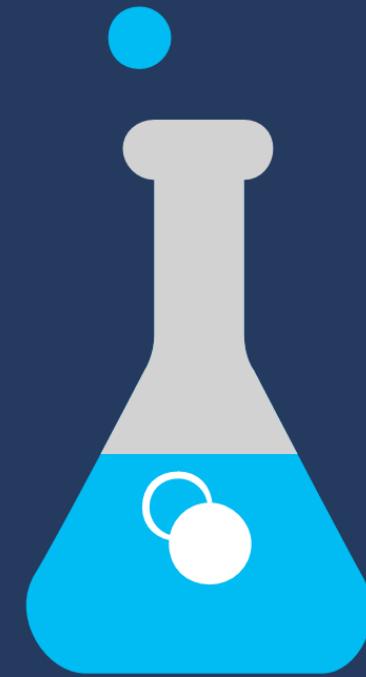
Calculations (optional)

Disconnected Tables  
(optional)



# Power BI Desktop

- Extending the Data Model



LAB

**Questions?**



# Knowledge Check

What are some of the differences between Calculated Columns and Calculated Measures?

Why should you avoid Auto Datetime feature?

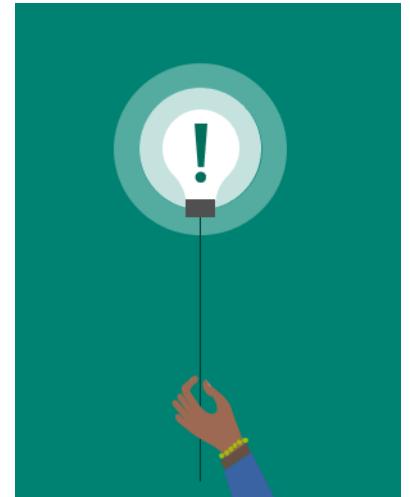
What are the advantages of using DAX variables?

# Lesson 6: Building a Report

# Objectives

After completing this learning, you will be able to:

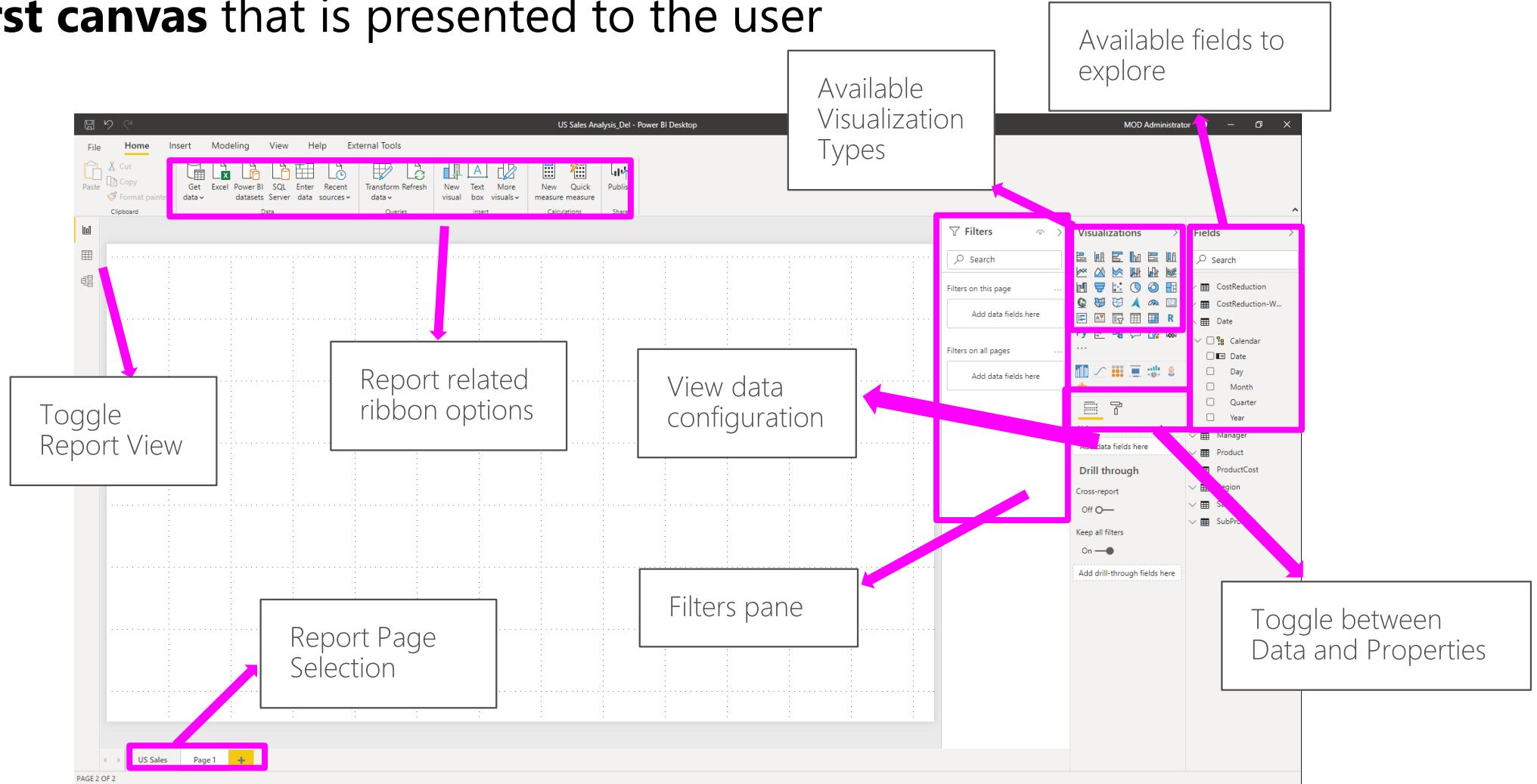
- Describe building a report
- Learn different chart types on Power BI Desktop
- Learn different interactive features on Power BI Desktop



# Building a Report

## Report View

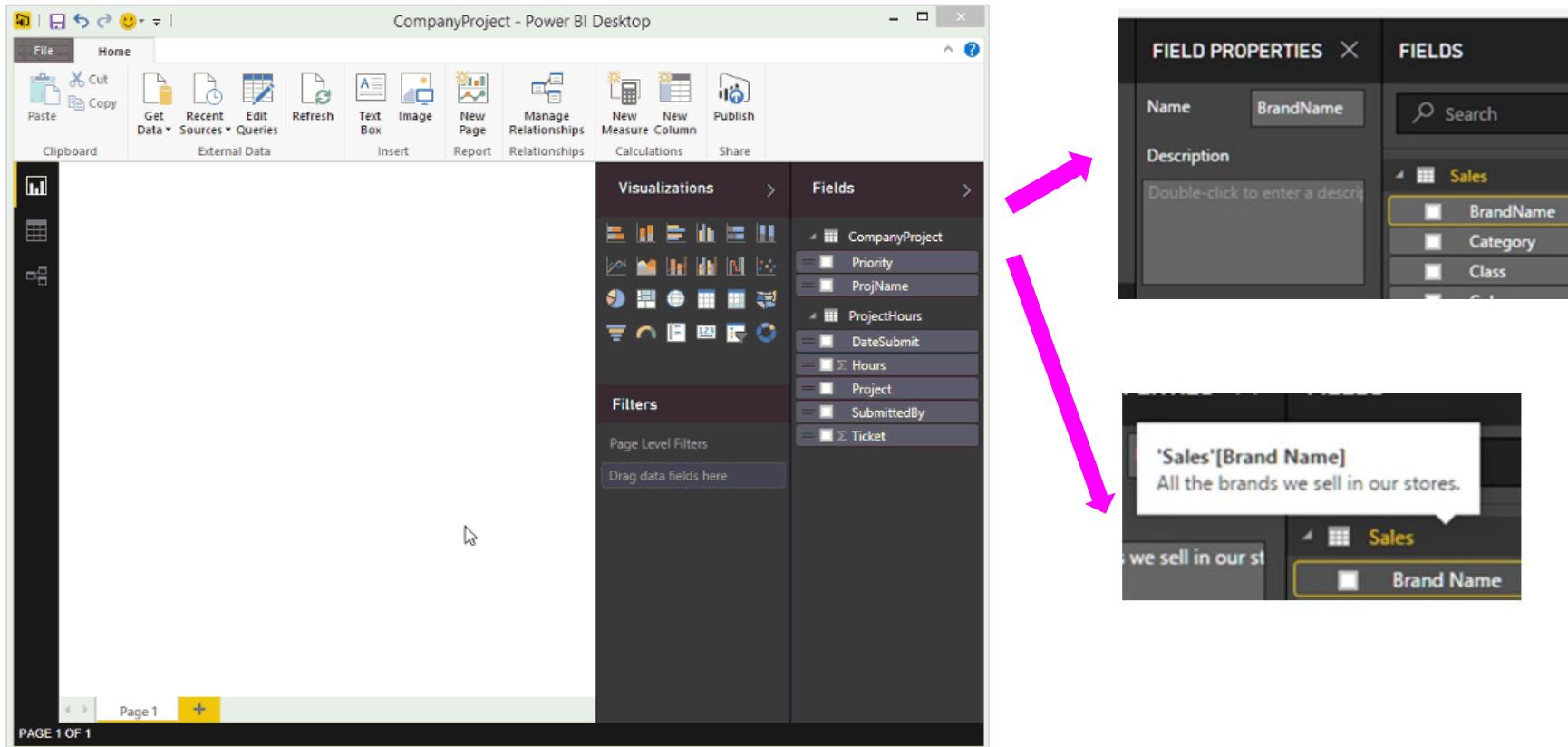
- The **first canvas** that is presented to the user



# Building a Report

## Report View

Use **drag-and-drop** and **search** to build your analysis like a pivot table in Excel

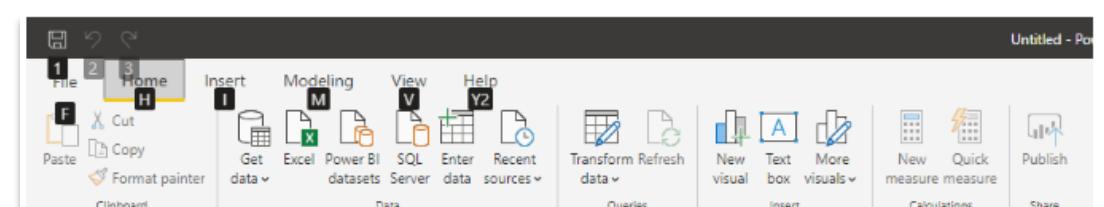
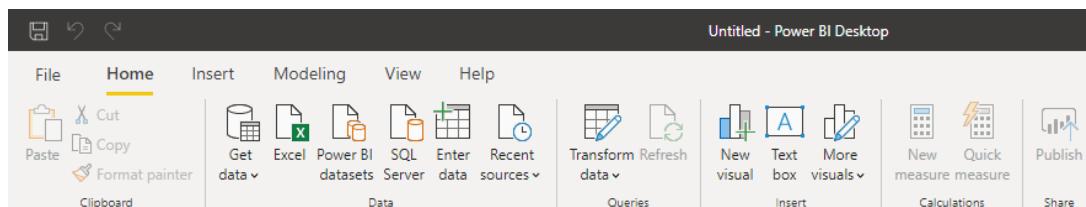


# Building a Report

## Modern Ribbon

### Modern Ribbon: same ribbon used for Office 365

- Improved look and feel and organization
- A themes gallery, where you can more easily see what colors will be applied
- Dynamic ribbon content based on your view, so you no longer have many disabled buttons in the data and modeling view
- A single line ribbon when the ribbon is collapsed, which allows you to save space while working
- The ribbon won't occasionally freeze, for those who had that issue with the previous ribbon
- You can use key tips to navigate and select buttons in the ribbon and title bar. To activate keytips, press Alt + Windows Key.



# Building a Report

## Keyboard Shortcuts

Click "?" to access keyboard shortcuts



Keyboard shortcuts

Across the product

<b>Ctrl+F6</b>	Move focus between sections
<b>Enter</b>	Select item in focus
<b>Esc</b>	Close out of current section
<b>Tab</b>	Move focus forward in section
<b>Shift+Tab</b>	Move focus backward in section
<b>?</b>	Show keyboard shortcuts

On visual

<b>Alt+Shift+F10</b>	Move focus to visual menu
<b>Alt+Shift+F11</b>	Show data

**Close**

# Building a Report

## Visualizations

---

**Several visualizations exist** out-of-the box and developed by 3rd party

---

Choose the one that **best conveys your message**

---

New visualizations are **released frequently**

---

**Change easily** between visualizations

---

Each visualization has its **own data settings and properties**

---

**Fine control on display settings**

---

**Visualizations can be responsive**

# Building a Report

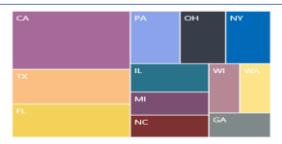
## Choosing the Right Charts for the Right Purpose

Comparison



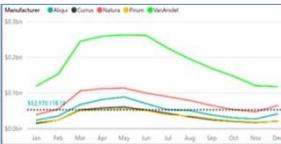
Stacked Bar

Composition



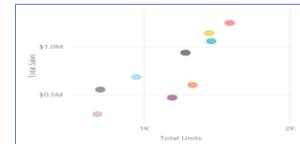
Tree

Trend



Line

Distribution



Scatter Plot

Status



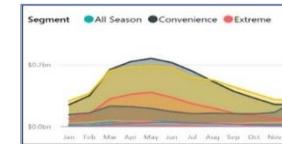
KPI

Geographic



Map

Relationships



Area

Slicing/Filtering

Date: 1/1/2011 - 12/31/2016

Region: Central, East, West

Segment: All

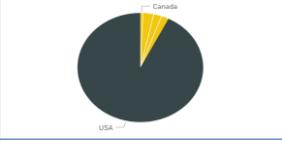
Product: Maximus RP-01, Maximus RP-02, Maximus RS-01, Maximus UC-00, Maximus UC-01, Maximus UC-02, Maximus UC-03, Maximus UC-04, Maximus UC-05, Maximus UC-06, Maximus UC-07, Maximus UC-08

Clustered Bar



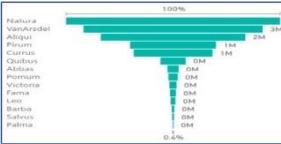
Clustered Bar

Pie



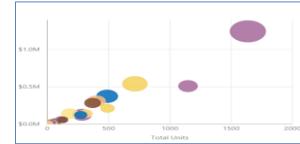
Pie

Funnel



Funnel

Bubble



Bubble

Gauge



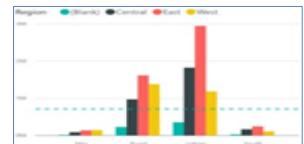
Gauge

Filled Map

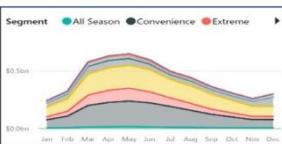


Filled Map

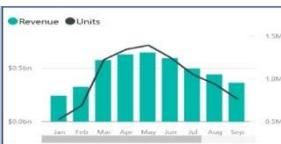
Bubble



Clustered Column



Stacked Area



Line and Stacked Column

Map



Map

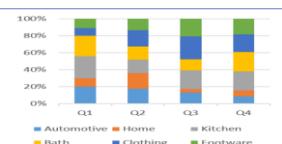
Multi-row Card



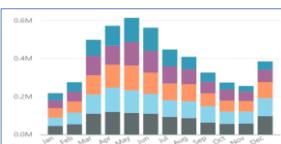
Multi-row Card

Table

Segment	Region	Units	Gross Profit
Moderation	Central	137,999	\$6,525,172
Convenience	Central	67,391	
Accessory	Central	15,028	
Productivity	Central	12,443	
All Season	Central	6,155	
Youth	Central	5,849	
<b>Total</b>	<b>Central</b>	<b>244,865</b>	<b>\$6,525,172</b>



100% Stacked Column



Stacked Column



Card



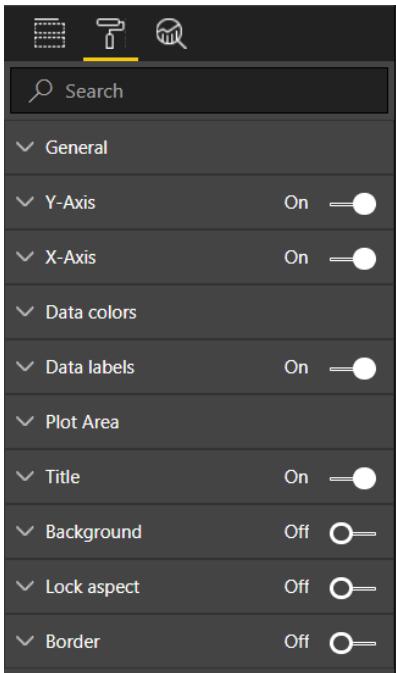
Matrix



Water Fall

# Building a Report

## Visualizations – Properties



- ➡ Filtering settings via text search
- ➡ Pixel level control of placement and size
- ➡ Enables the X/Y-Axis, control alignment, title and color and scale (Y - linear/log)
- ➡ Enables a reference line for a particular value and with formatting options
- ➡ Granular control on the color of each axis occurrence
- ➡ Display the values associated to a series. Control the unit, color and font
- ➡ Plot area formatting, namely define an image as background
- ➡ Enable the title and format it
- ➡ Enable a background color and define transparency
- ➡ Lock the current aspect
- ➡ Enable a border and control its color

\* Properties vary, depending on visualization.

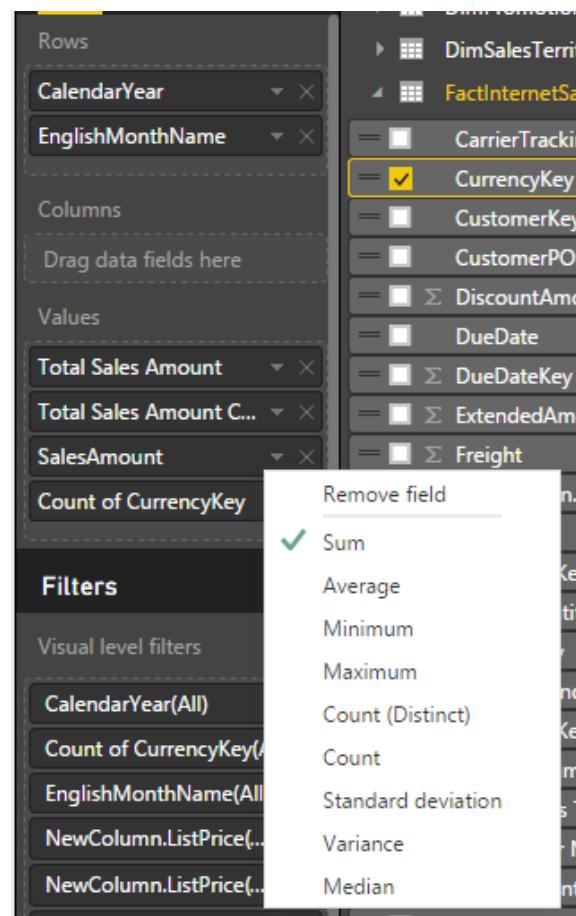
# Building a Report

Common visual properties - Numerical fields and aggregations

Any **numerical field** can have a summarization other than its default

A numerical field with default “**Do not summarize**” **can still be aggregated** in the field pane

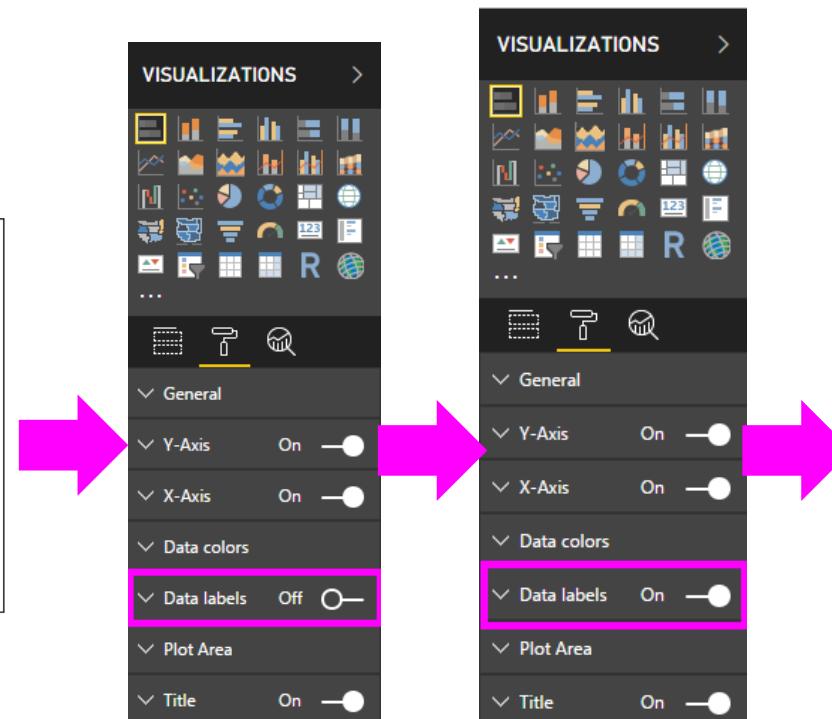
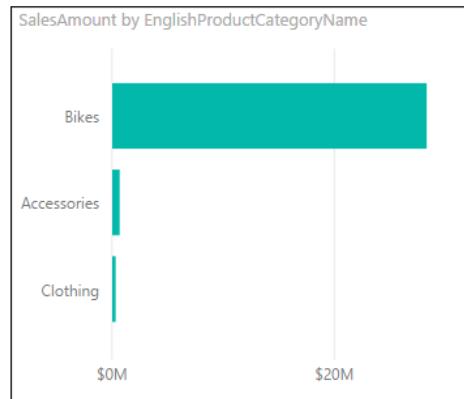
**Measures cannot change** their aggregation



# Building a Report

## Common visual properties - Data Labels

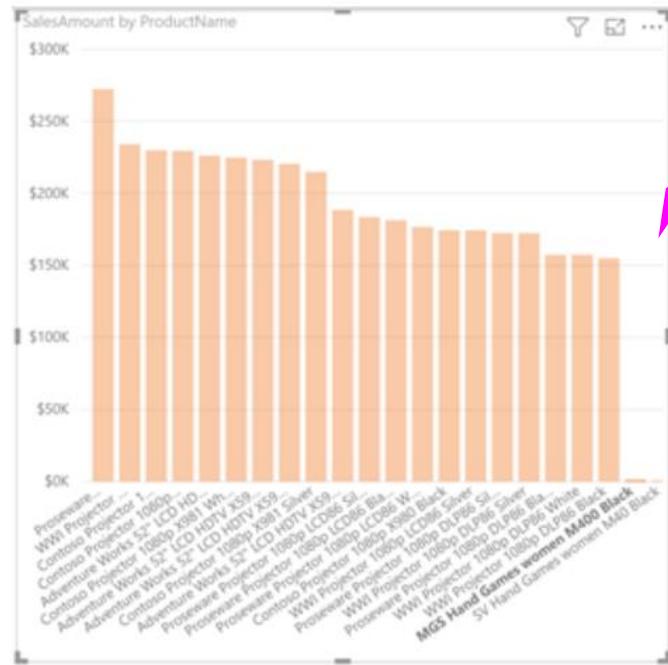
**Data labels** make a chart easier to understand, because they show details about a data series or its individual data points.



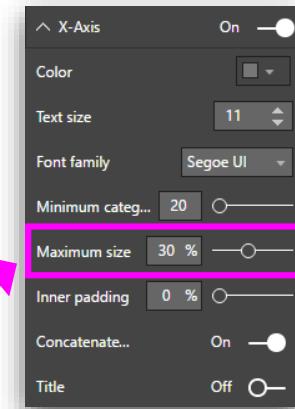
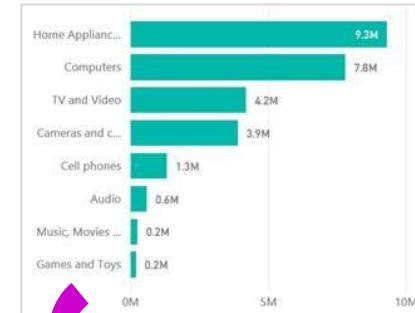
# Building a Report

Common visual properties - Bar chart controls

Increase area used for axis labels & Cross highlight axis labels



**Maximum size** control Slider can be used to increase/decrease the percentage of the chart used by the **axis labels..**



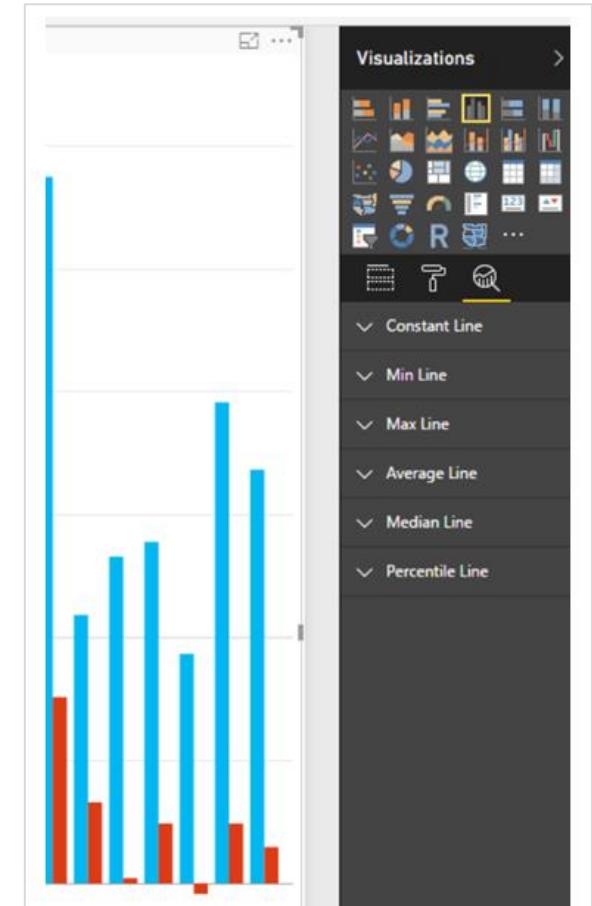
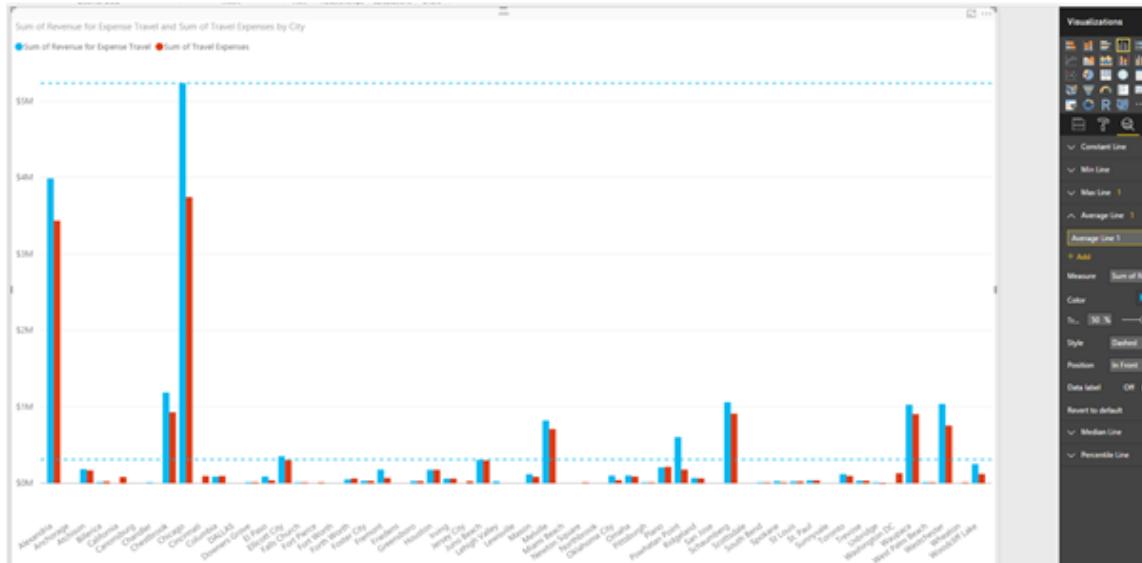
# Building a Report

Common visual properties - Analytics pane

**Analytics pane** will be the central location for all analytical features

Add dynamic reference lines and trend lines on charts

Analytical features available only for **select visuals**



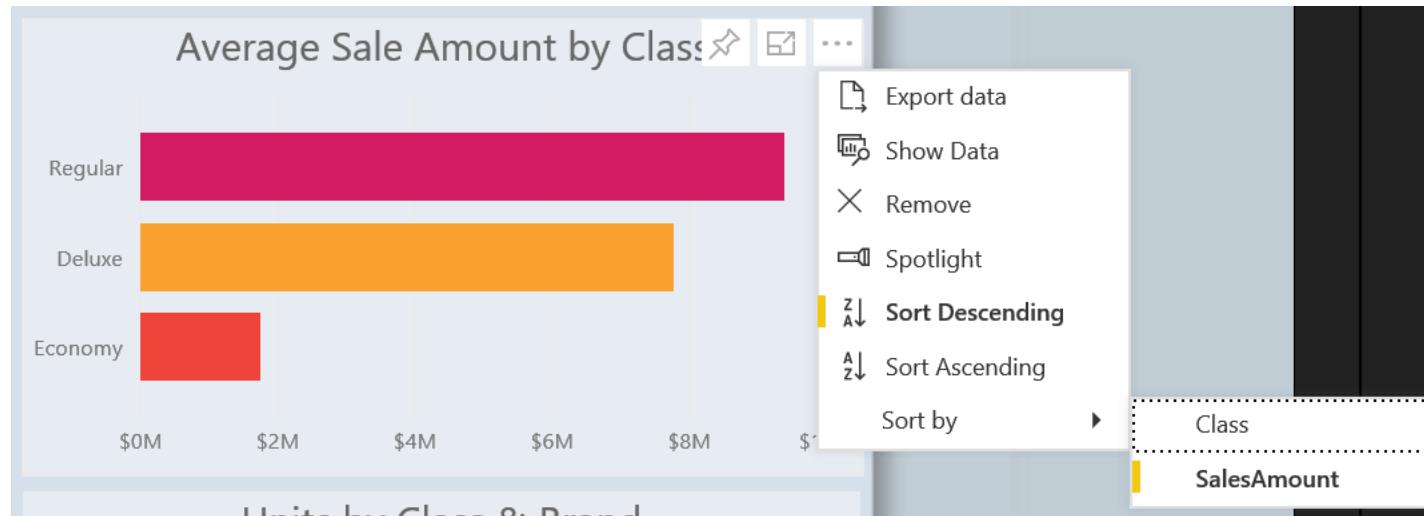
# Building a Report

## Common visual features - Sorting

It is possible to **sort** data according to **numeric values** or to **data categories**, for each **visualization**

For categories, the order that is used is either **alphabetical** (or **numerical**) or the **custom sort** order

Data can be presented **ascending** or **descending**



# Building a Report

Common visual features – Export, Focus mode and more

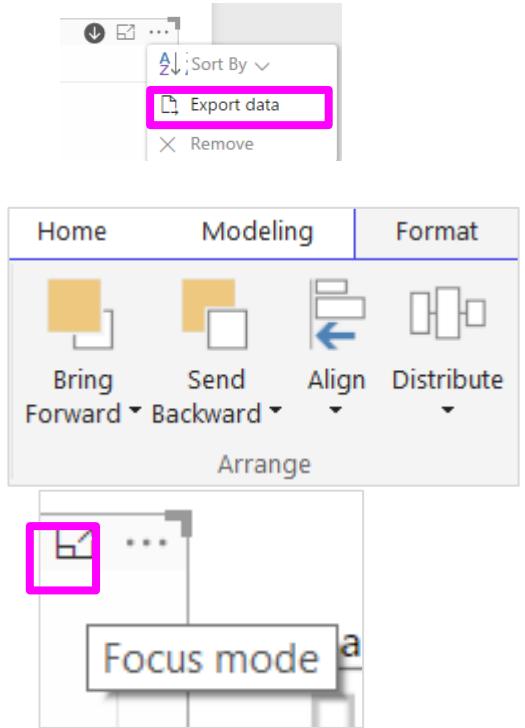
Data for a visualization can be **exported** as .csv file

**Alignment** and **Depth** can be controlled

**Focus mode** pops-up the visualization

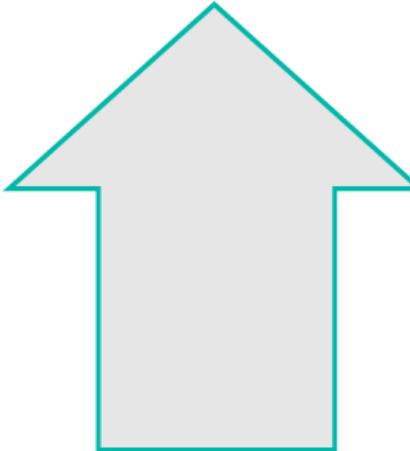
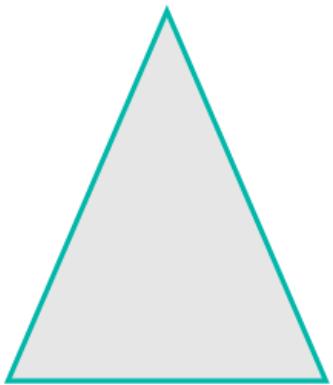
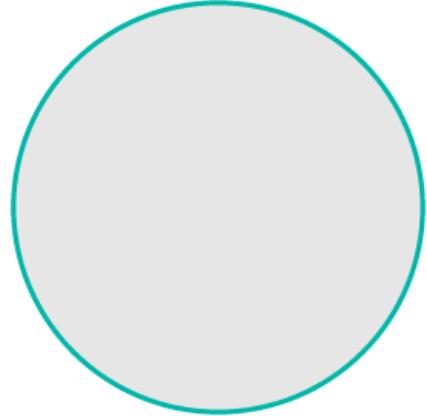
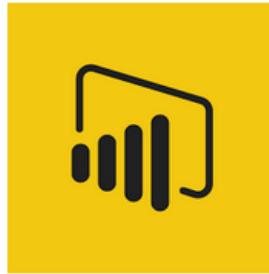
It is possible to **overlay** visualizations

It is possible to **copy visuals between files** (schema must match!)



# Building a Report

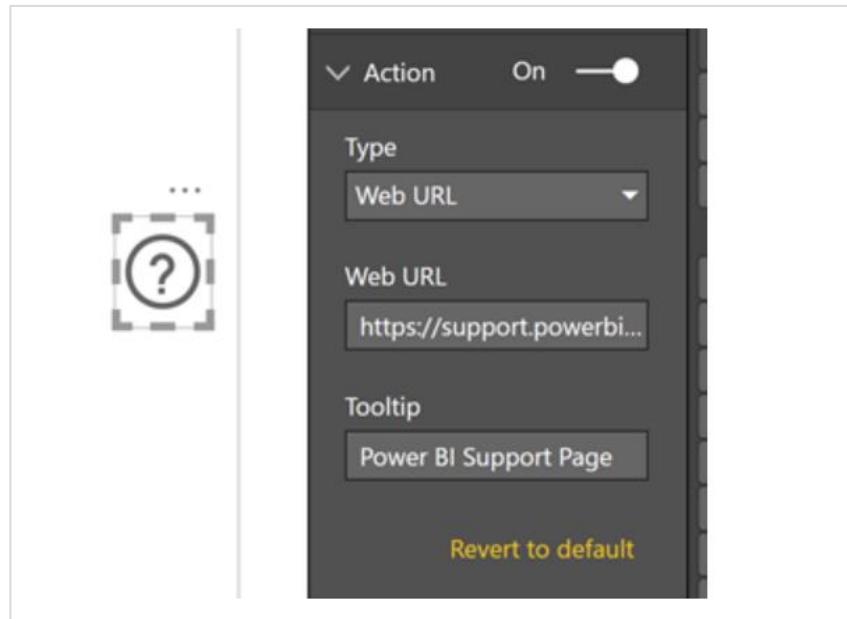
Visualizations – Textboxes, Images and Shapes



# Building a Report

Static web URL support for buttons, shapes, and & images

You can link a **specific website** through an **image, shape** or **button**. You can do this by selecting the **Web URL option** in the **Type dropdown** of the Action card of the formatting pane.



# Building a Report

## Wallpapers

With new wallpaper feature, we can format the grey area outside report page to theme the entire report area.

If the page background has a transparency higher than 50%, you will see a grey dotted line in editing mode to help see where the edge of your page is.



# Building a Report

## Report Themes

Definition of **theme as JSON file**

Define your **custom color palette**

Define your **default value of settings**

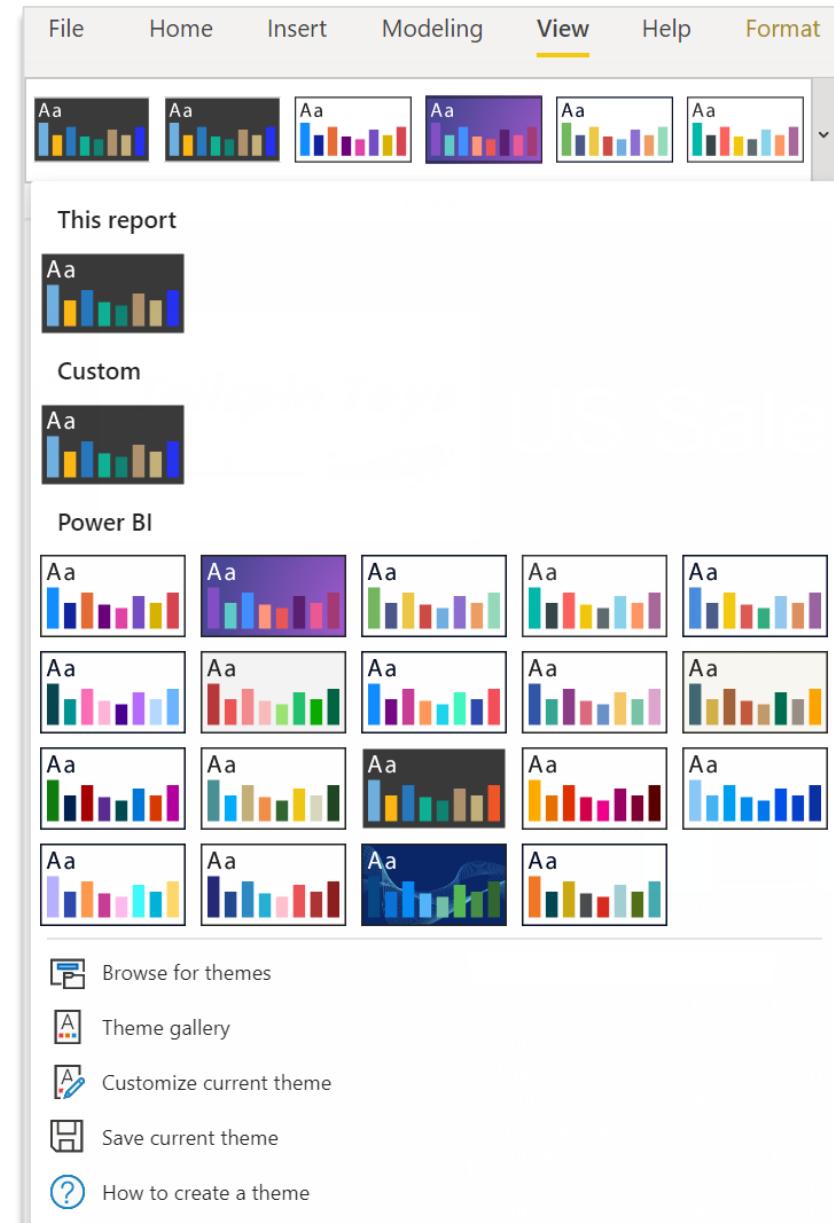
**Built-in** report themes available

The color palette will be **applied to all visualizations** (except if custom data points were defined)

The theme gets applied in the service, too.

There are **generators** themes, you don't need to code!

Possible to customize through a dialog



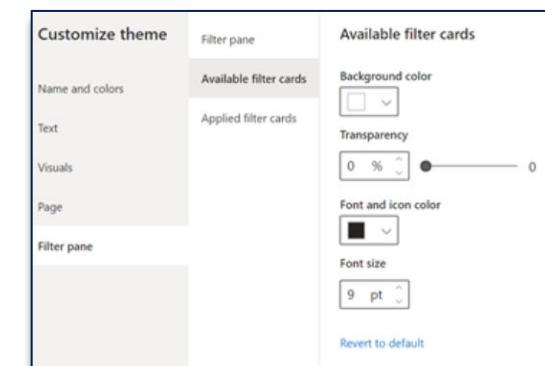
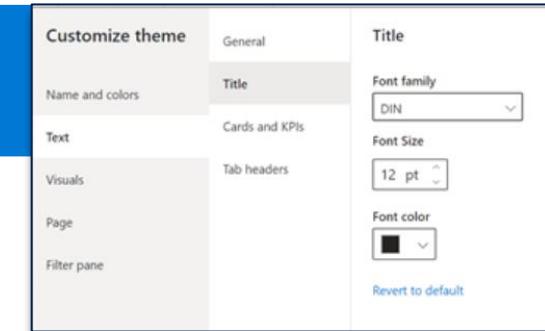
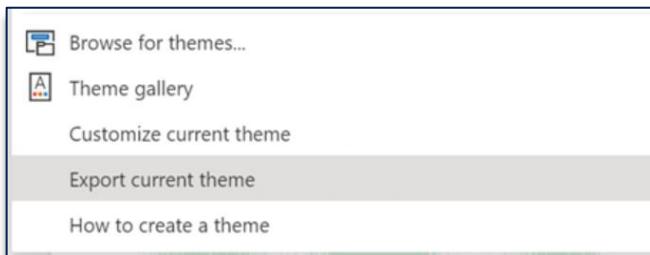
# Building a Report

## Customize & Export Current Theme

The most common options of your current theme can be **customized through a dialog** in Power BI Desktop instead of needing to write or modify a JSON file.

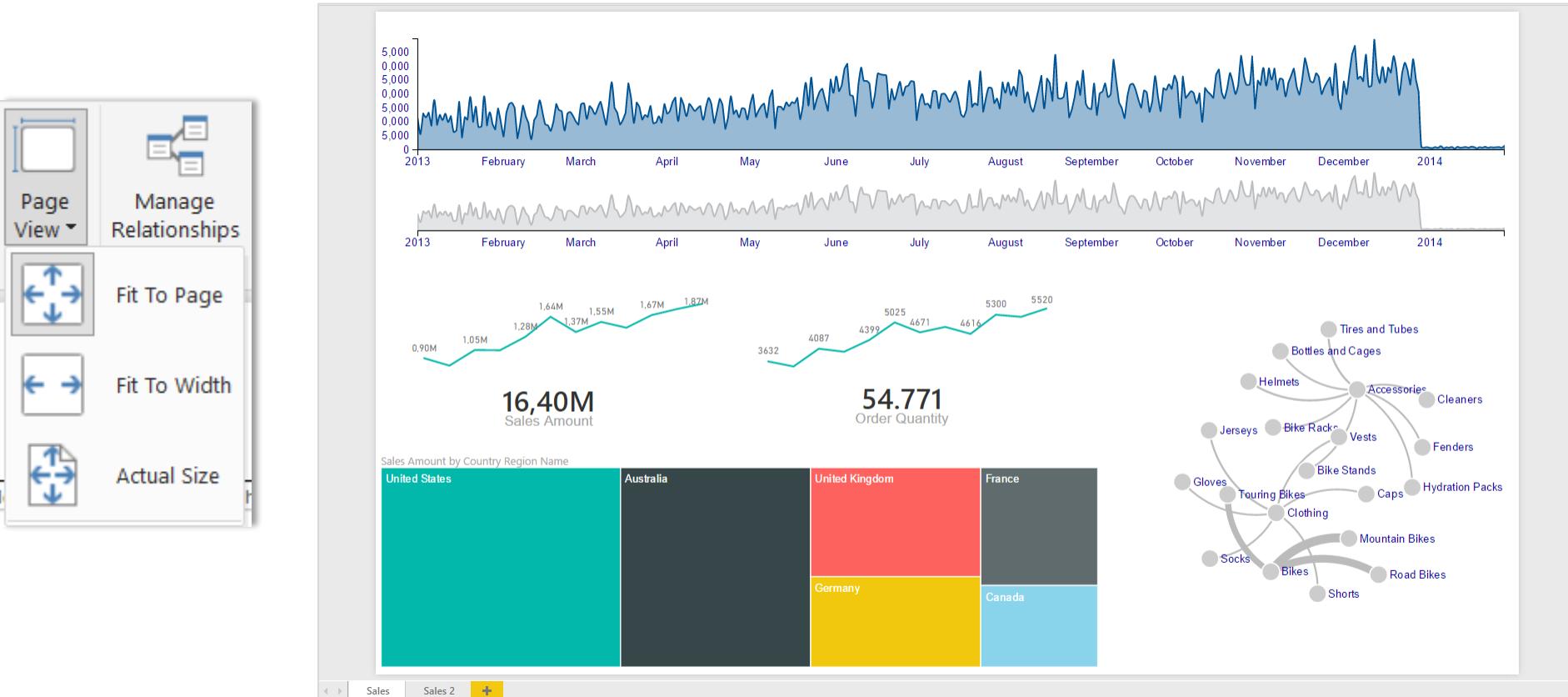
The settings that can be customized are divided into **different categories** like **name and colors, text, visuals, page settings**

Custom theme can also be **exported** to be reused



# Building a Report

## Page View



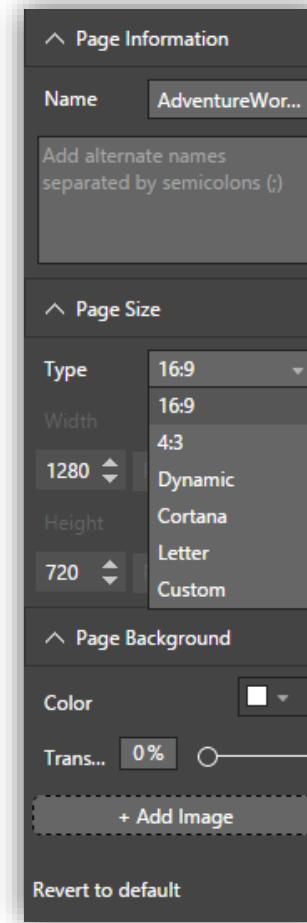
# Building a Report

## Page Properties

Name

Page Size and Format

Page Background  
(both color and image)



# Building a Report

## Hidden pages

Right click on **Page** and select “**Hide Page**” to hide it in **Reading View**

Hidden Pages still shown on Design View, but **look different to indicate they are hidden**

Hidden pages can still be accessed through **drill through or bookmark**

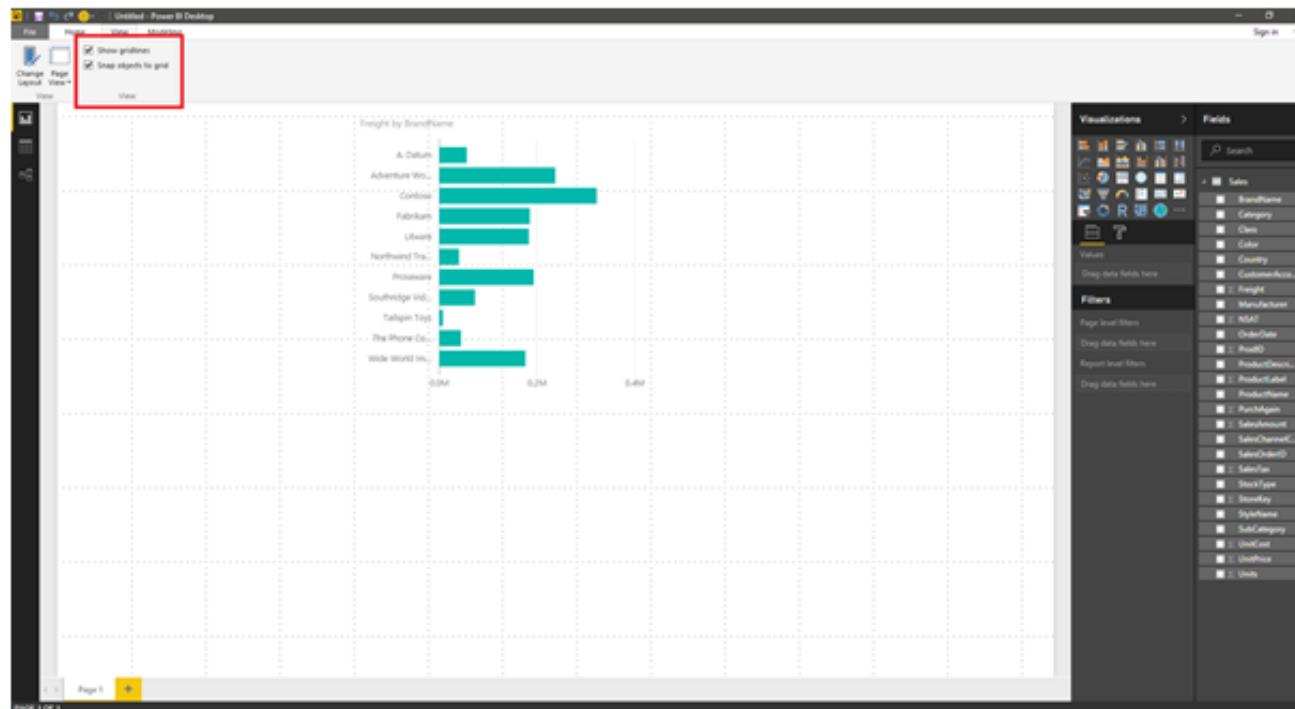


# Building a Report

Report gridlines and snap to grid

**Enable gridlines** on your report canvas while authoring

**Snap to gridlines**



# Demonstration

## Report Design Basics



# Building a Report

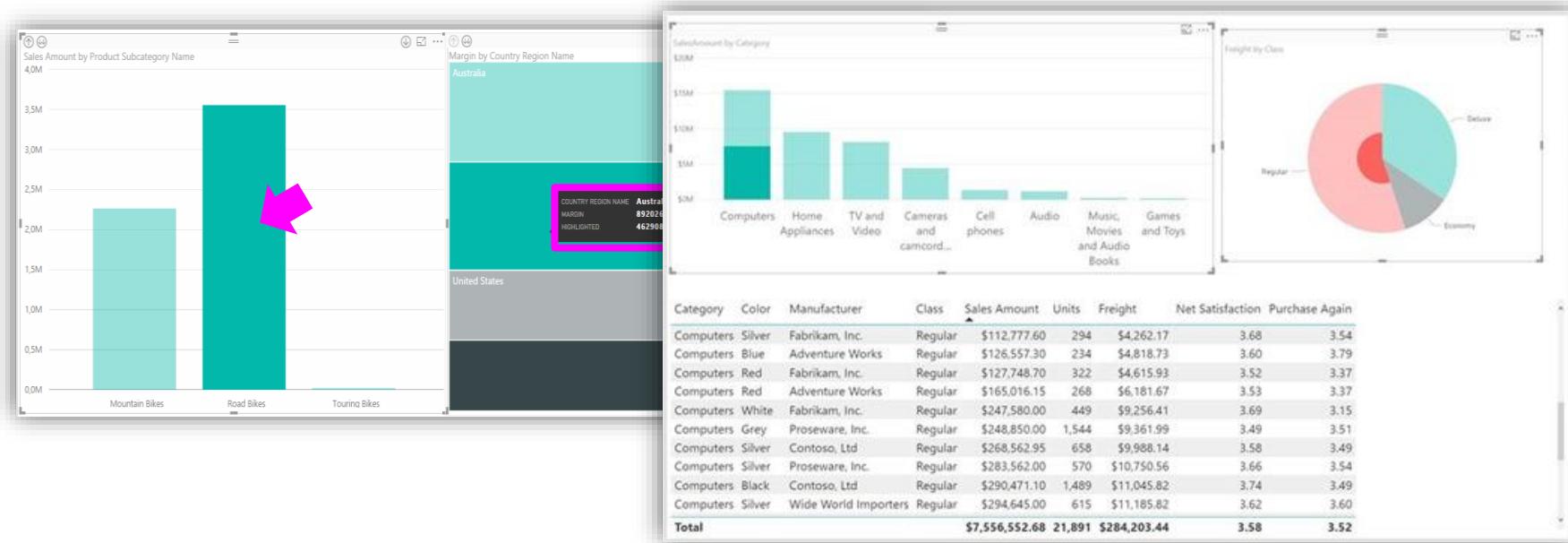
Interactive Features - Cross-Filtering and Cross-Highlight

**Clicking or drilling on a visualization element, filters other visualizations**

You can **multi-select data points** across multiple visualizations

A **tooltip** displays the **highlighted portion** and the total

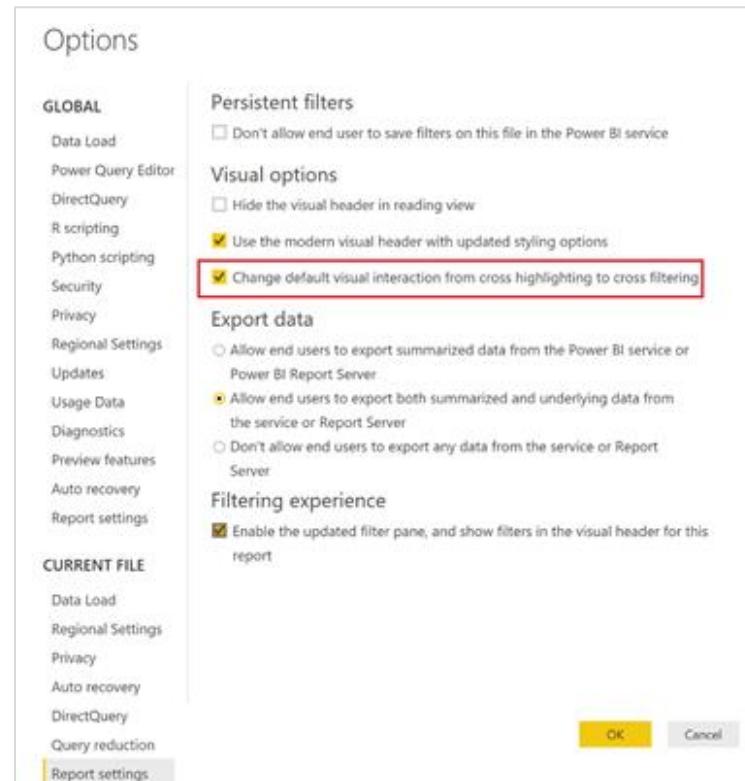
Click an **empty space** to clear the filter



# Building a Report

## Interactive Features - Cross-Filtering and Cross-Highlight

Can change default visual interaction from cross highlighting to cross-filtering

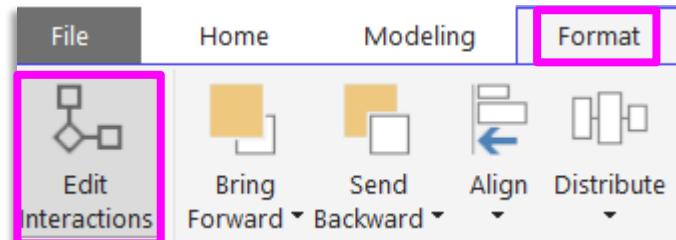
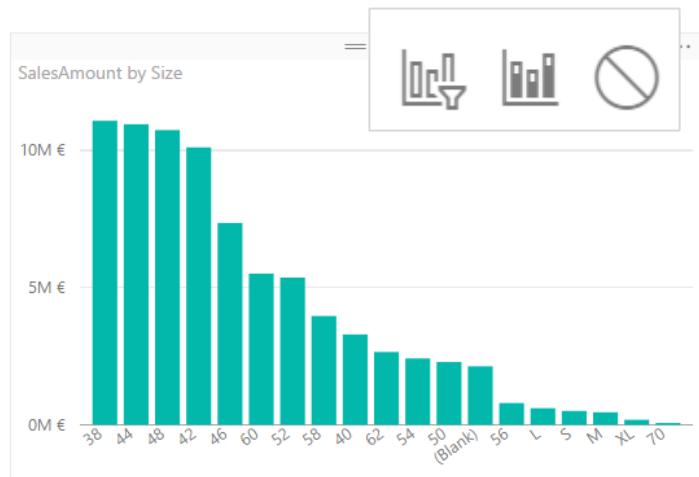


# Building a Report

## Interactive Features - Visual Interactions

You can control interactions between visualizations

Click a visualization, the master, and on the other visualizations select the type of interactivity



This option enables cross-filter



This option enables cross-highlight



This option disables filtering

# Building a Report

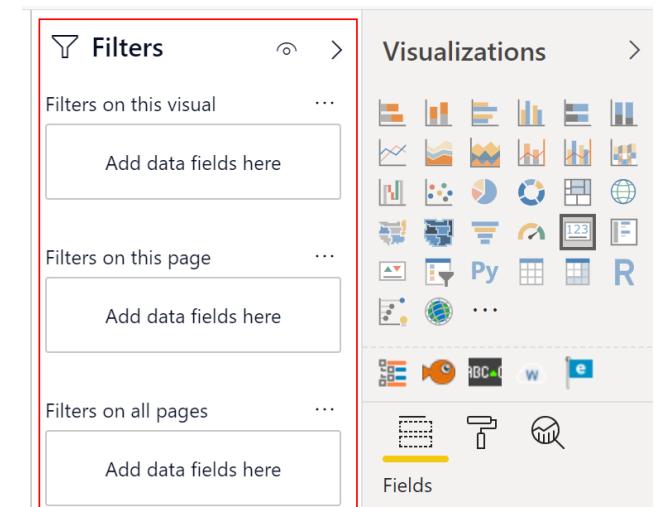
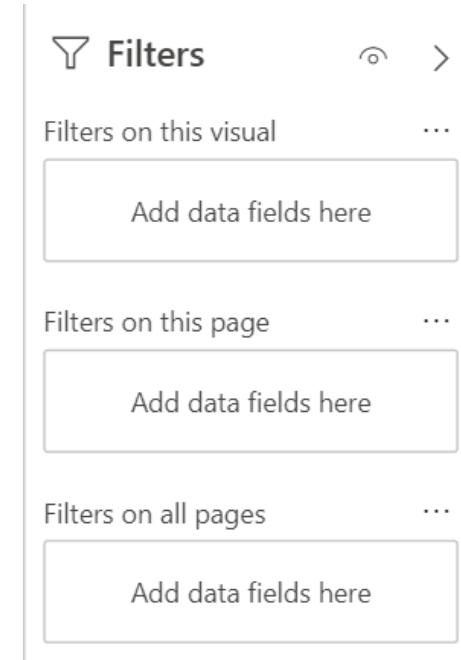
## Interactive Features - Report Filtering

Filters can be applied at three different levels:

- **Visual** – The filter only applies to the current visualization
- **Page** – The filter applies for every visualization on the page
- **Report** – The filter applies for every visualization for every page in the report (file)

Filter experience

- Dedicated filter pane with advanced options.



# Building a Report

## Interactive Features - Report Filtering

### Filtering pane configuration and customization features:

**Add/Remove** individual filters

**Show/Hide individual filters** or the entire filter pane from report consumers

**Lock filters** as read only for report consumers

**Formatting options** for the filter pane to make it feel like part of the report

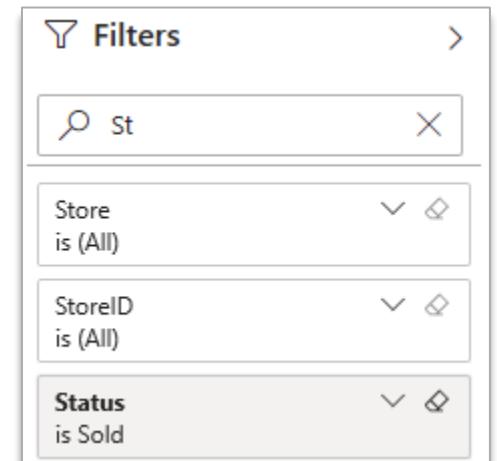
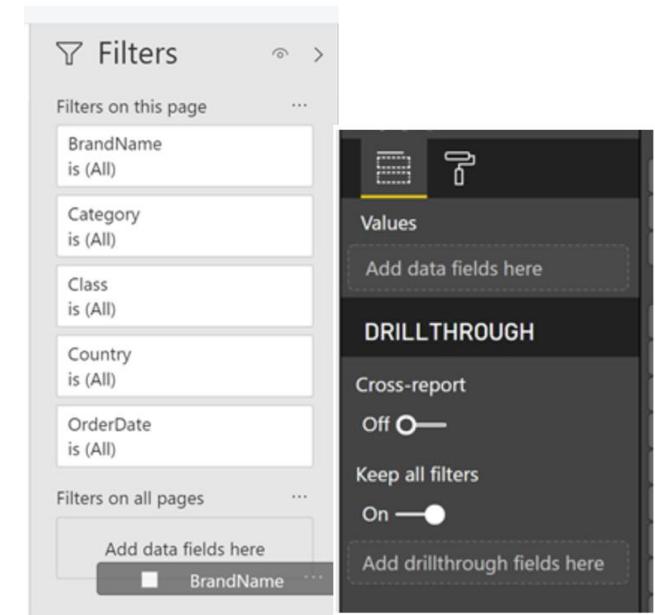
Flexibility of **defining default** expand / collapsed filter pane at report load time for consumers.

**Rename** filters by double clicking the filter title.

**Sort** the filter pane

**Filter pane search** allows you to search across your filter cards.

Supports **basic, advanced, conditional (AND OR), Top N, relative date filters and more**



# Building a Report

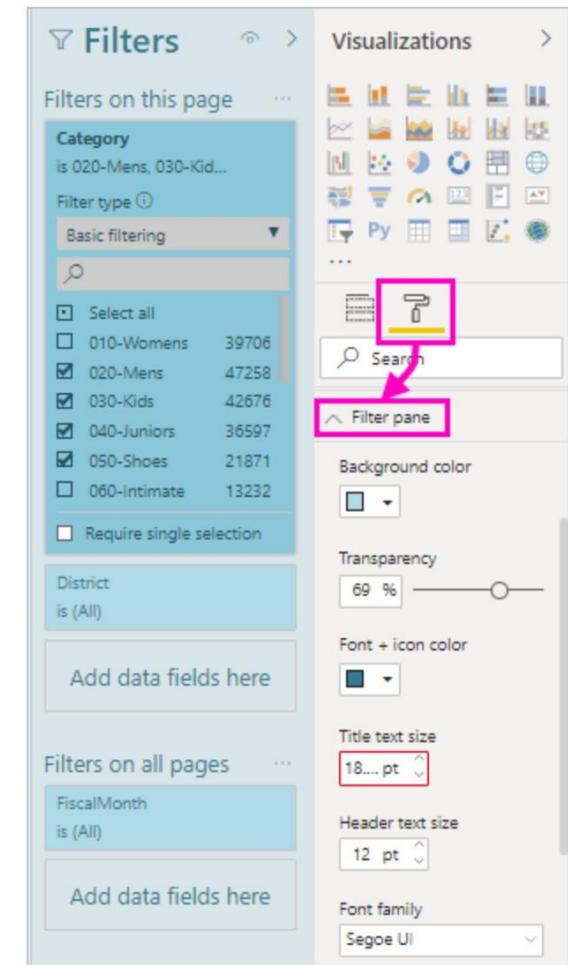
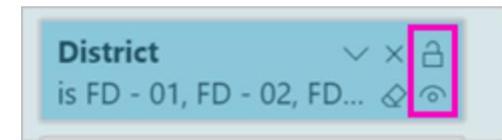
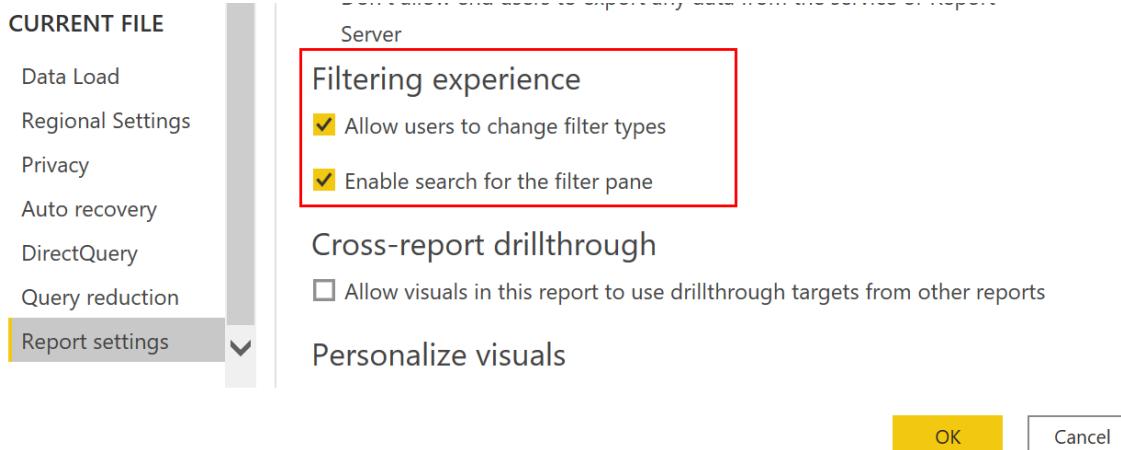
## Interactive Features - Report Filtering

### Filtering pane configuration and customization features (cont.):

Lock or **hide** individual filter cards.

Use the formatting pane to **format the entire pane as well as each filter card** in the pane. Can format using theme files too.

**Report Settings -> Filtering experience** section, includes option to restrict changes to filter type. If turned off, **report consumers won't have access to the dropdown to switch between the basic and advanced types of filters**.



# Building a Report

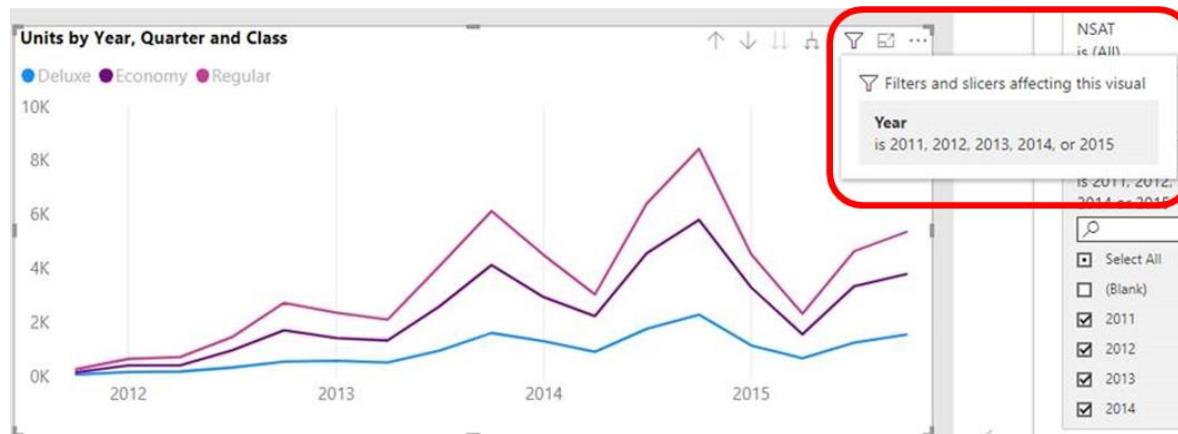
## Report Filtering Pane (cont.)

### Filtering pane configuration and customization features (cont.):

Report consumers can see a read **only view of filters that are affecting** the visual.

**Hover over a filter icon** in the visual header to see the filters, slicers, cross-highlighting state that is affecting the data on the visuals.

Can **turn on** this new filter icon through filter **formatting options**.



# Building a Report

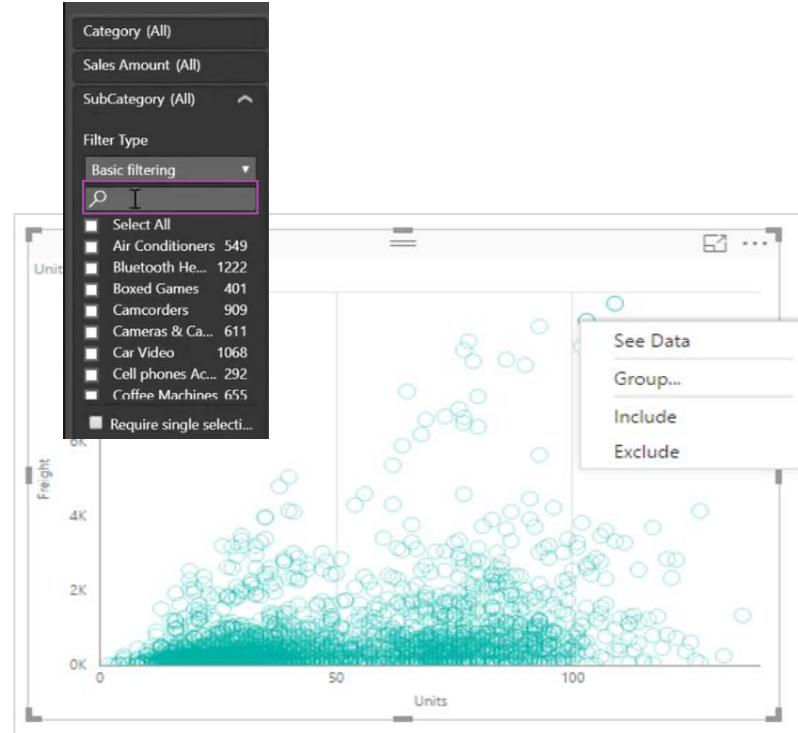
## Interactive Features - Report Filtering

**Basic filtering**, which allows **searching**

**Filters on measures** differ in behavior when placed at different filtering levels

- **Visual Level** – the filter is for the **aggregated** values (default is sum)
- **Page or Report Level** – the filter is applied at **row level**

**Include/Exclude** data points



# Building a Report

## Interactive Features - Slicers

**Filter** data on **multiple visualizations** (like Excel slicers)

Allow **Select All**

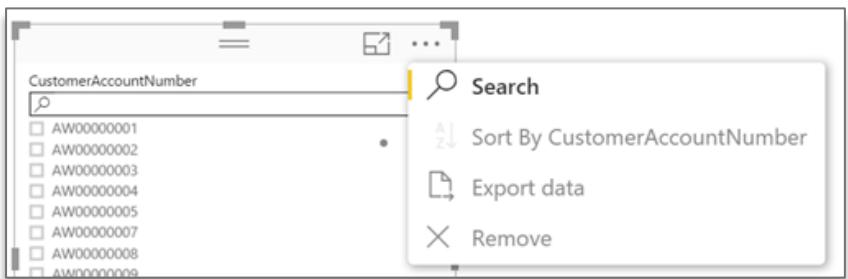
**Multi** and **Single** selection

Radio buttons and allows **one and only one filter selection**

Special **slicers** for date field (relative!) and numeric

Slicers are **searchable**

**Hierarchical slicer** with various collapse/expand icons



A screenshot of a Power BI report interface. It features several interactive elements:

- A top navigation bar with month names: January, February, March, April, May (highlighted in black), June, and a right arrow.
- An 'Order Date' slicer with a date range selector set to '7/15/2010' to '3/20/2013' and a 'Between' dropdown.
- A 'CalendarYear' slicer with a list of years from 2005 to 2014, where 2008 and 2010 are checked.
- A 'Date' slicer with a date range selector set to 'Last' (1) 'Years' (1) and a date range of '13-06-2016 - 12-06-2017'.
- A 'Brand Name' slicer on the right with a list of brand names: A. Datum, Contoso, Fabrikam, Litware, Northwind Traders, Proseware, Southridge Video, Tailspin Toys, The Phone Company, and Wide World Importers.
- A 'Country' slicer with a list of countries: Australia, Canada, France, Germany, Italy, Spain, and the United Kingdom.
- A 'Visu...' button in the bottom right corner of the visualization pane.

A screenshot of a hierarchical slicer interface. The categories listed are:

- Category, Subcategory, Brand Name
  - Audio
  - Cameras and camcorders
    - Camcorders
    - Cameras & Camcorders Acces...
    - Digital Cameras
  - Digital SLR Cameras
    - A. Datum
    - Contoso
    - Fabrikam
  - Cell phones
  - Computers

A screenshot of a 'Slicer header' configuration pane. It includes:

- Search input field.
- General category.
- Selection controls category.
- Slicer header switch: 'On' (selected).
- Title text input field: 'Type, Platform, Name'.

# Building a Report

## Interactive Features - Slicers

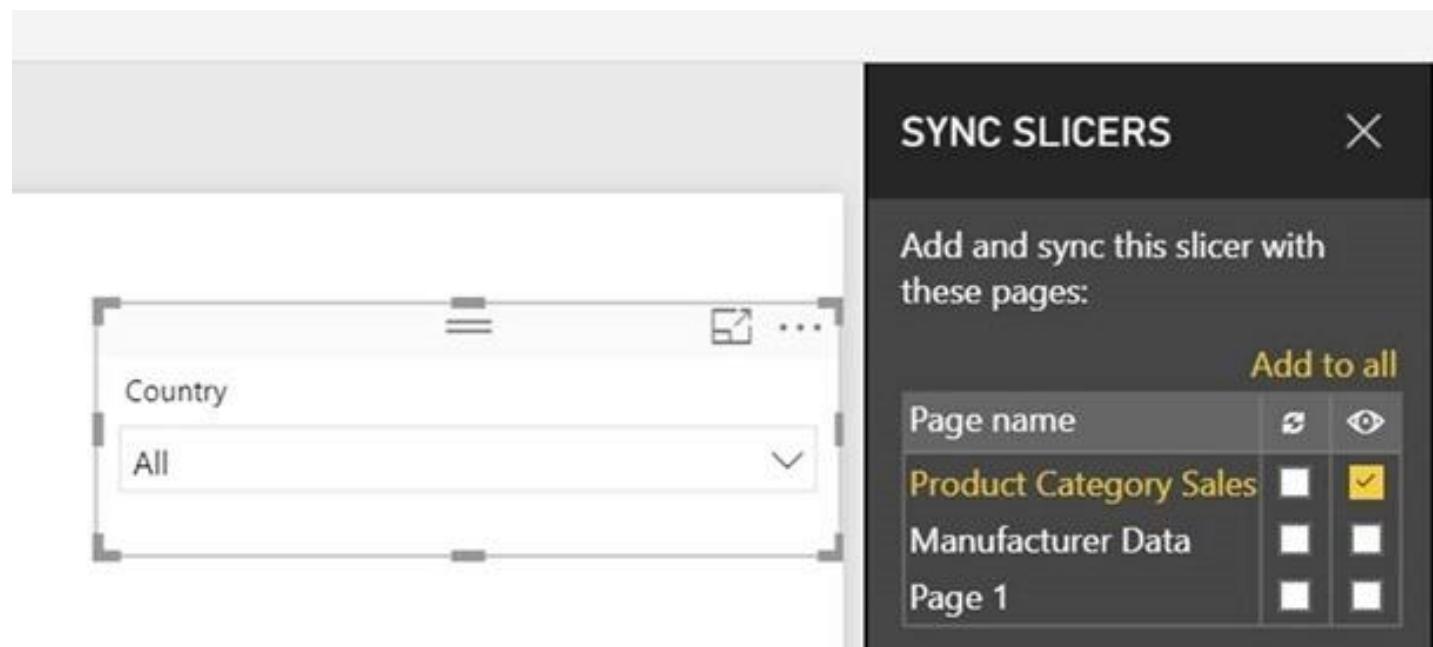
Sync slicers allows the applications of filters to multiple pages

You can set this behavior through the "Sync slicers" pane

Hidden slicer will be put on every page, and you can control visibility

Create custom groups of slicers to sync under Advanced options

Supports standard OOB slicer visual and some custom visual like Hierarchy slicer



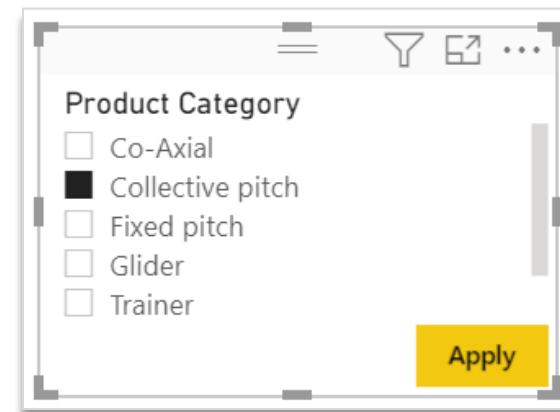
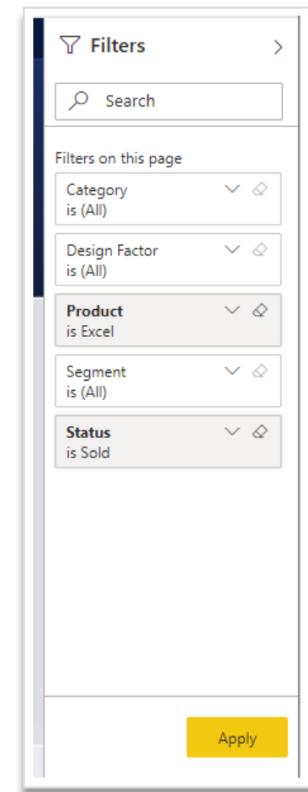
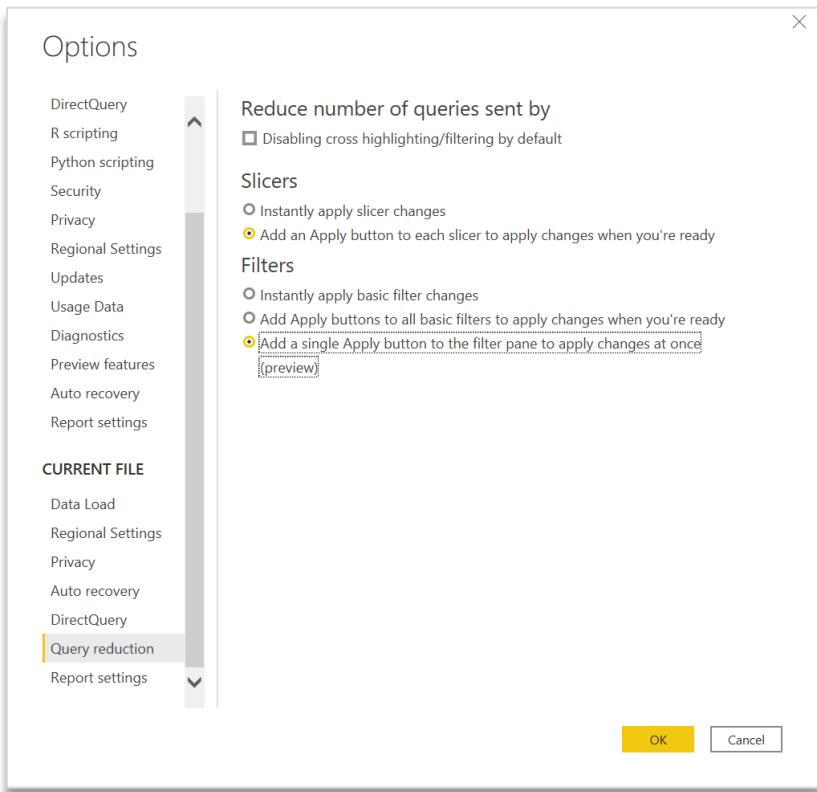
# Building a Report

## Query reduction

You can **reduce the number of queries** when you use slicers or filters

It is useful for **very large** or **very slow data sources** in DirectQuery

Set to enable an **"Apply"** button that will appear in the objects

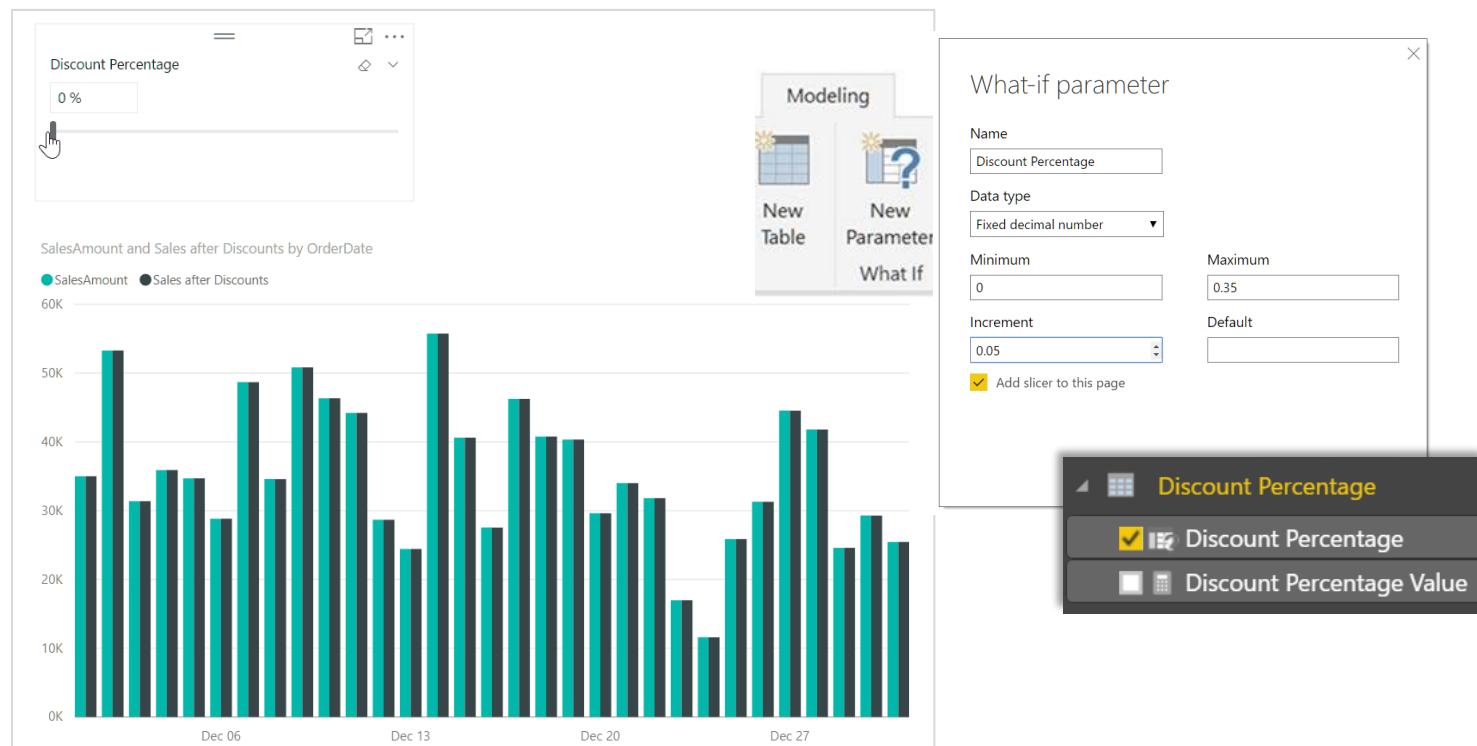


# Building a Report

## Interactive Features - What-If Parameter

Create new parameters and use it for **what-if analysis**

You have to apply it in your own calculations



# Demonstration

**Visual Interactions, Filters & Slicers**



# Building a Report

## Visualizations – Table and Matrix

Table contains related data in a logical **series of rows and columns**; may also contain headers and a row for totals

Matrix is like a table but automatically **aggregates the data and enables drill down**, making it easier to display data meaningfully across multiple dimensions

Contains **predefined styles** for Table and Matrix

Category	This Year Sales Status	Average Unit Price	Last Year Sales	This Year Sales	This Year Sales Goal
010-Womens	●	\$7.30	\$2,680,662	\$1,787,958	\$2,680,662
020-Mens	●	\$7.12	\$4,453,133	\$4,452,421	\$4,453,133
030-Kids	●	\$5.30	\$2,726,892	\$2,705,490	\$2,726,892
040-Juniors	●	\$7.00	\$3,105,550	\$2,930,385	\$3,105,550
050-Shoes	●	\$13.84	\$3,640,471	\$3,574,900	\$3,640,471
060-Intimate	●	\$4.28	\$955,370	\$852,329	\$955,370
070-Hosiery	●	\$3.69	\$573,604	\$486,106	\$573,604
080-Accessories	●	\$4.84	\$1,273,096	\$1,379,259	\$1,273,096
090-Home	●	\$3.93	\$2,913,647	\$3,053,326	\$2,913,647
100-Groceries	●	\$1.47	\$810,176	\$829,776	\$810,176
Total	●	\$5.49	\$23,132,601	\$22,051,952	\$23,132,601

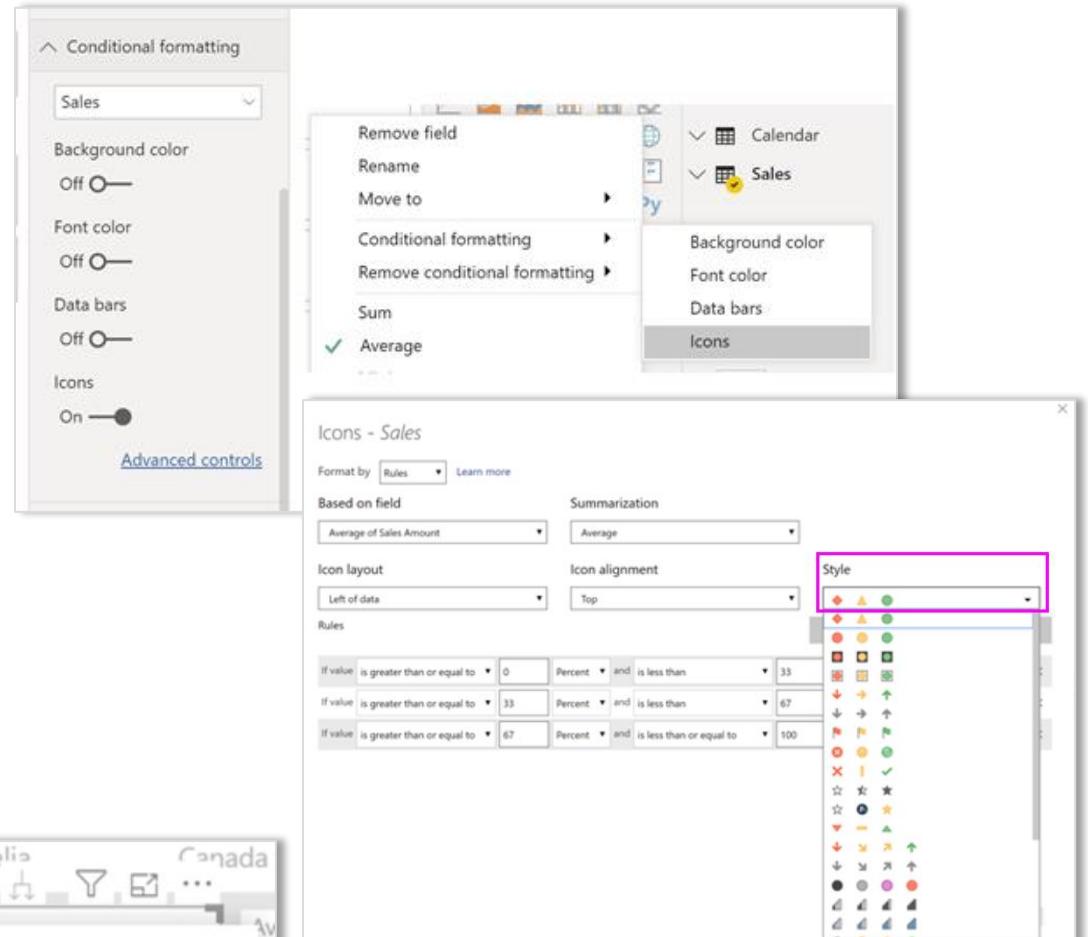
Region	East			West			Total	
	Sales Stage	Opportunity Count	Revenue	Opportunity Count	Revenue	Opportunity Count	Revenue	Opportunity Count
Lead		114	\$473,887,837	52	\$256,159,114	166	\$730,046,951	
Small		44	\$55,009,105	13	\$11,979,738	57	\$66,988,843	
Medium		31	\$120,678,637	18	\$72,871,697	49	\$193,550,334	
Large		39	\$298,200,095	21	\$171,307,679	60	\$469,507,774	
Proposal		13	\$59,970,924	10	\$43,032,669	23	\$103,003,593	
Small		3	\$4,770,862	3	\$3,720,287	6	\$8,491,149	
Medium		6	\$25,607,581	5	\$21,456,937	11	\$47,064,518	
Large		4	\$29,592,481	2	\$17,855,445	6	\$47,447,926	
Finalize		5	\$30,696,428	4	\$21,176,185	9	\$51,872,613	
Small		1	\$1,693,585			1	\$1,693,585	
Medium				2	\$7,926,517	2	\$7,926,517	
Large		4	\$29,002,843	2	\$13,249,668	6	\$42,252,511	
Total		132	\$564,555,189	66	\$320,367,968	198	\$884,923,157	

# Building a Report

Visualizations – Table and Matrix (Conditional Formatting)

## Conditional formatting options

- **Background color**
- **Font color**
- **Data bars**
- **Icons**

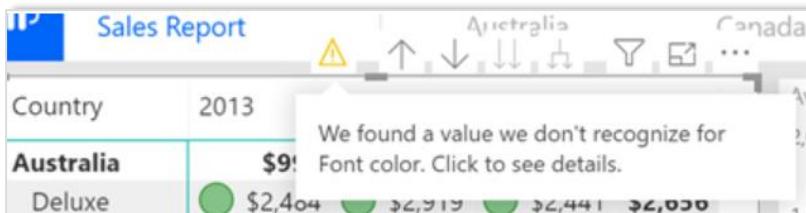


The screenshot shows the Power BI desktop application with the 'Conditional formatting' pane open for the 'Sales' field. The pane lists options like Background color, Font color, Data bars, and Icons. The 'Icons' option is selected. A blue arrow points from the 'Icons' section of the list above to this pane. To the right, a detailed view of the 'Icons - Sales' dialog box is shown, containing three rules based on average sales amount:

Rule	Condition	Icon
1	If value is greater than or equal to 0	Green circle
2	If value is greater than or equal to 33	Yellow triangle
3	If value is greater than or equal to 67	Red square

## Conditional Formatting Warnings

- Invalid measure data type
- Deleted measure
- Unrecognized measure



Country	2013	Australia					Canada		
Australia	\$9!								
Deluxe	\$2,404						\$2,919	\$2,441	\$2,030

We found a value we don't recognize for Font color. Click to see details.

# Building a Report

## Visualizations – Table and Matrix (Conditional Formatting)

The screenshot illustrates the process of applying conditional formatting to data visualizations.

**Left Panel (Conditional Formatting Options):**

- Icon bar: Includes icons for Home, Conditional formatting (highlighted with a red box), and Filter.
- Search bar: "Search" icon.
- Section: "Conditional formatting" (highlighted with a red box).
- Rule: "Sum of Net Revenue" (dropdown menu).
- Background color: "On" slider.
- Font color: "Off" slider.
- Data bars: "Off" slider.
- Icons: "Off" slider.
- Web URL: "Off" slider.

**Middle Panel (Conditional Formatting Rule Editor):**

Two instances of the "Background color scales" dialog box are shown, connected by pink arrows:

- Top Dialog:** "Base value" is "Net Profit". "Format blank values" is "As zero". "Color by rules" is checked. "Diverging" color scale is selected, ranging from red (Lowest value) to green (Highest value). Buttons: "OK" and "Cancel".
- Bottom Dialog:** "Base value" is "Net Profit". "Color by rules" is checked. "Rules" section shows three conditions:
  - If value is greater than or equal to Minimum and is less than 100000, then red.
  - If value is greater than or equal to 100000 and is less than 1000000, then yellow.
  - If value is greater than or equal to 1000000 and is less than 10000000, then green.Buttons: "OK" and "Cancel".

**Right Panel (Resulting Visualizations):**

**Top Visualization:** A table showing SalesAmount by Category and SubCategory, with conditional formatting applied to the SalesAmount column. The colors range from red (low values) to green (high values). Total values are also present.

Category	SubCategory	SalesAmount
Audio	Bluetooth Headphones	378,685.86
	MP4&MP3	482,111.29
	REconomyrd Pen	304,390.85
<b>Total</b>	<b>1,165,188.00</b>	
Cameras and camcorders	Camcorders	2,048,841.00
	Cameras & Camcorders Accessories	161,040.01
	Digital Cameras	1,079,453.90
	Digital SLR Cameras	1,167,697.10
<b>Total</b>	<b>4,457,032.01</b>	
Cell phones	Cell phones Accessories	18,704.00
	Home & Office Phones	76,156.95
	Smart phones & PDAs	745,239.00
	Touch Screen Phones	518,090.00
<b>Total</b>	<b>1,358,189.95</b>	
Computers	Computers Accessories	699,162.40
	Desktops	1,692,976.80
	Laptops	3,771,848.75
	Monitors	2,155,865.00
	Printers, Scanners & Fax	1,219,935.00
	Projectors & Screens	5,946,536.00

**Bottom Visualization:** A table showing Net Profit by Product Category and Product Name, with conditional formatting applied to the Net Profit column. The colors range from red (low values) to green (high values).

Product Category	Product Name	Net Profit
Co-Axial	3CAX-B Helicopter	256,033
	4CAX-B Helicopter	270,166
Co-Axial	Tailspin Heli - Co-Ax Pro Mk I - 4ch	631,918
	6CCP-A Helicopter	2,094,360
Collective pitch	Tailspin Heli - Max Pro Flight - 6ch	8,353,028
	Fixed pitch	4CFP-I Helicopter
Fixed pitch	4CFP-I Helicopter	14,610
	Tailspin Heli - Pro Mk III - 5ch	15,392
Glider	Tailspin Aviator Mk2-11	113,469
	Glider	Trainer - Tailspin GL-120
Glider	Trainer - Tailspin GL-155	90,878
	Trainer	Piper Cub 3 Channel
Trainer	Piper Cub 4 Channel	1,245,153
	SkyTrainer	193,035
Trainer	Tailspin Aviator Mk2-12	615,820
	Tailspin Aviator Mk2-15	1,026,410
Warbird	P47 4 Channel	130,678
	P47 5 Channel	1,010,973

*Conditional formatting can also be applied to a column **based on a different field***

# Building a Report

## Visualizations – Table and Matrix

Supports copying a **specific value** or a **selection of data** from a table or matrix

Copy value will add the **unformatted value** to your clipboard

Copy selection will add a **tabular version of all data currently selected** in your table or matrix to the clipboard, including the row and column headers

	SalesAmount	ProductName
Year	Delux	
2015	\$290.1K	SV USB Data Cable E600
Qtr 4	\$128.5K	SB Data Cable E600
Qtr 3	\$67.2K	SB Data Cable E600
Qtr 1	\$56.1K	Wire 80mm Dual Ball
Qtr 2	\$38.4K	SV H
2014	\$370.5K	Copy selection
Qtr 4	\$178.1K	Copy value
Qtr 3	\$82.3K	Copy selection
Qtr 1	\$73.9K	Copy value
Qtr 2	\$36.1K	Copy selection
2013	\$94.7K	Copy value
Qtr 4	\$68.9K	Copy selection
Qtr 3	\$24.2K	Copy value
Qtr 2	\$7.5K	Copy selection
Qtr 1	\$1.6K	Copy value
2012	\$29.6K	Copy selection
Qtr 3	\$19.5K	Copy value
Qtr 4	\$9.0K	Copy selection
Qtr 1	\$1.5K	Copy value

# Building a Report

## Visualizations – Matrix

Like in Excel, it is possible to **drill on specific elements** of a matrix, whether on columns or rows

The screenshot displays a Power BI matrix visualization. The matrix has 'Category' as the primary dimension across the rows and 'Sales Amount' and 'Units' as measures across the columns. The data includes:

Category	Sales Amount	Units
Computers	\$15,486,324	46,213
Home Appliances	\$9,564,741	18,128
TV and Video	\$8,155,111	16,560
Cameras and camcorders	\$4,457,032	12,466
Cell phones	\$1,358,190	7,814
Audio	\$1,165,188	8,643
Music, Movies and Audio Books	\$257,625	2,358
Games and Toys	\$196,763	4,427
<b>Total</b>	<b>\$40,640,973</b>	<b>116,609</b>

A context menu is open over the 'Category' column for the 'Total' row. The menu items are:

- Expand
- Collapse
- Drill Down
- Show Next Level
- Expand to next level
- Show Data
- Include
- Exclude
- Copy
- Selection
- Entire level
- All

The 'Selection' item is highlighted. On the right side of the matrix, there is a ribbon of icons and settings for alignment, +/- icons, icon color, and icon size. The 'Icon size' is set to 10, and the 'On' icon is selected.

# Building a Report

Map visuals – Bubble map

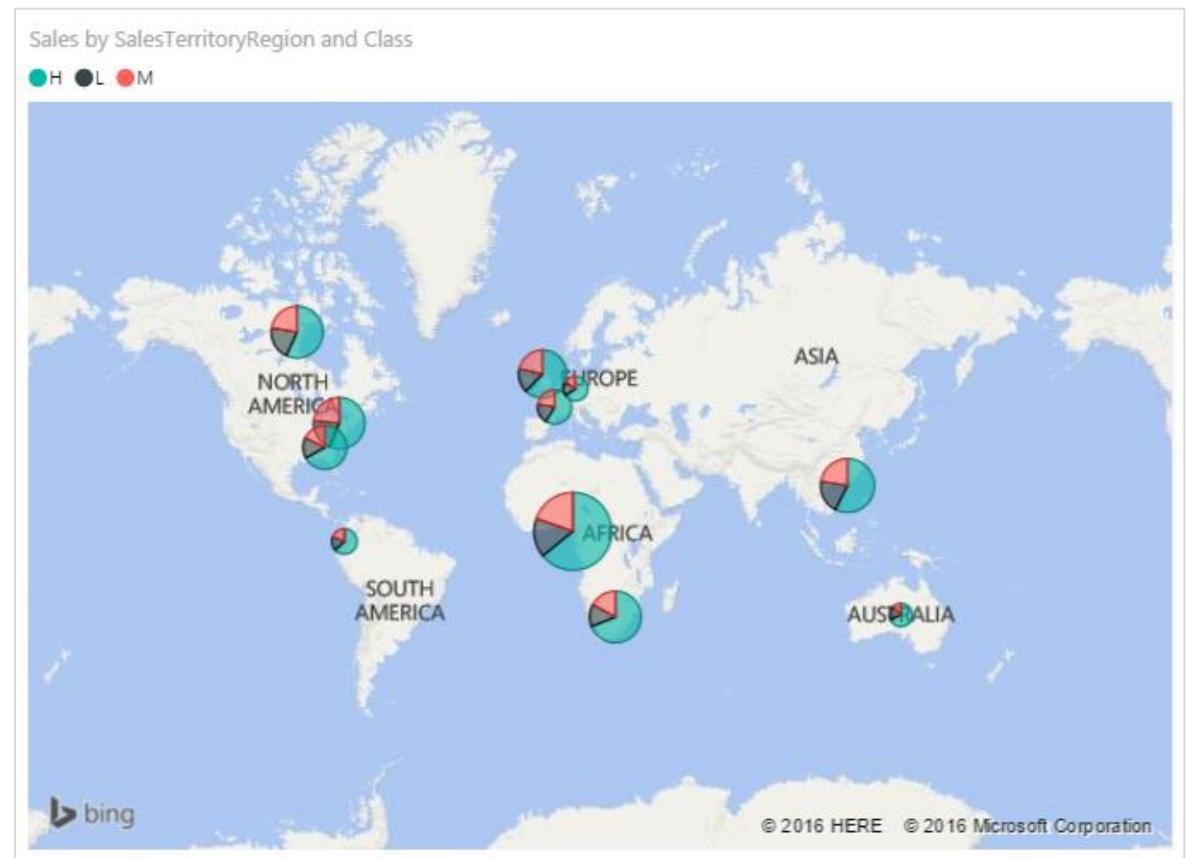
**Plotting data** requires internet connection

**Bing** is used to return the location

**Latitude and Longitude** can be used, but...

**Location names** are also mapped automatically

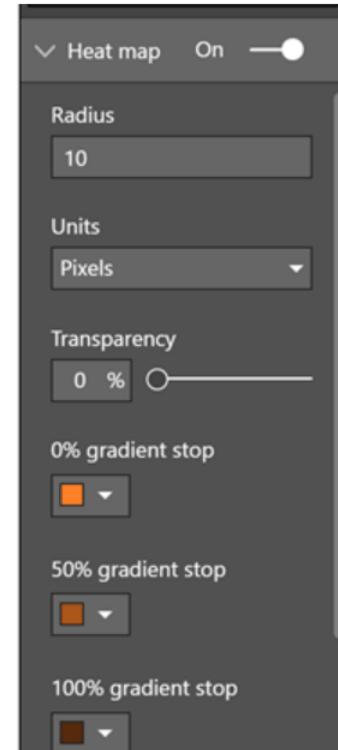
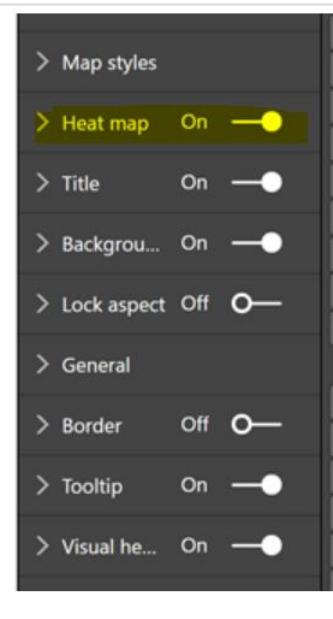
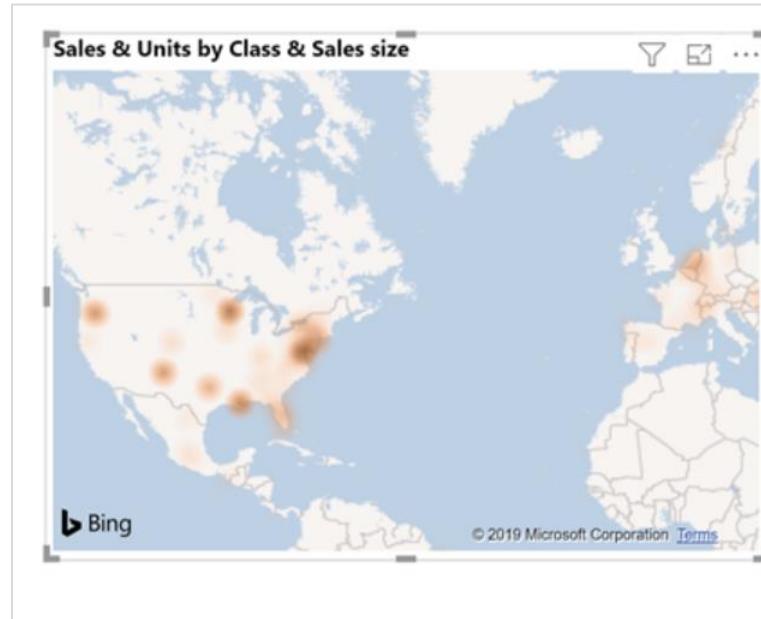
Combine attributes in Location for increased accuracy and use data categories



# Building a Report

## Map visuals – Heat map

**Heat** map formatting option for circle **Bing** map visual converting all the **location** dots on your map to a **heat map layer** through the formatting pane.



# Building a Report

## Map visuals – Filled map



# Building a Report

## Map visuals - ArcGIS

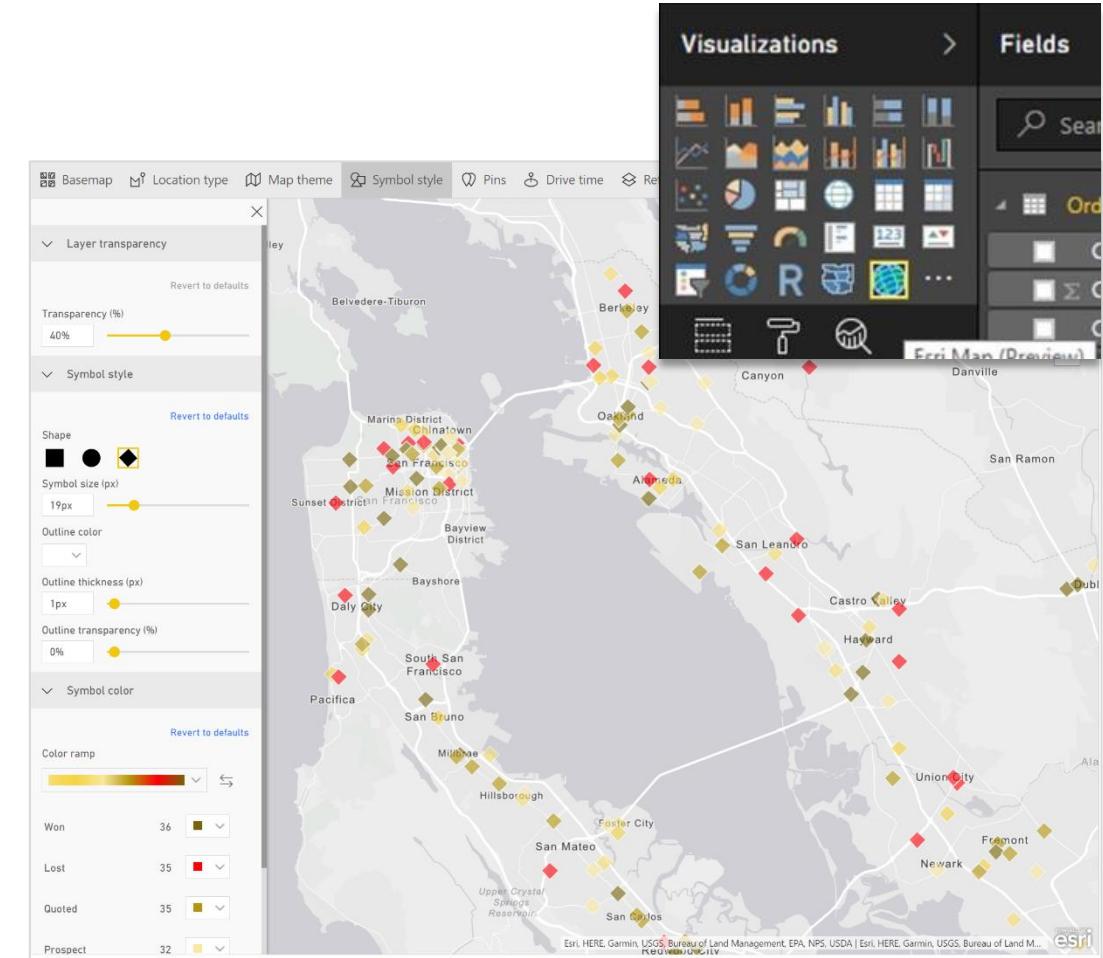
Use **ArcGIS** Maps as a Power BI visual

Different base maps (dark gray, light gray, open street maps, and Esri's standard ArcGIS base map)

Different themes: Points, Points with Size; Heat Map and Clusters

Additional Reference Layers (**demographic, public** layers)

Location is adaptive: streets are points, counties are boundaries



# Building a Report

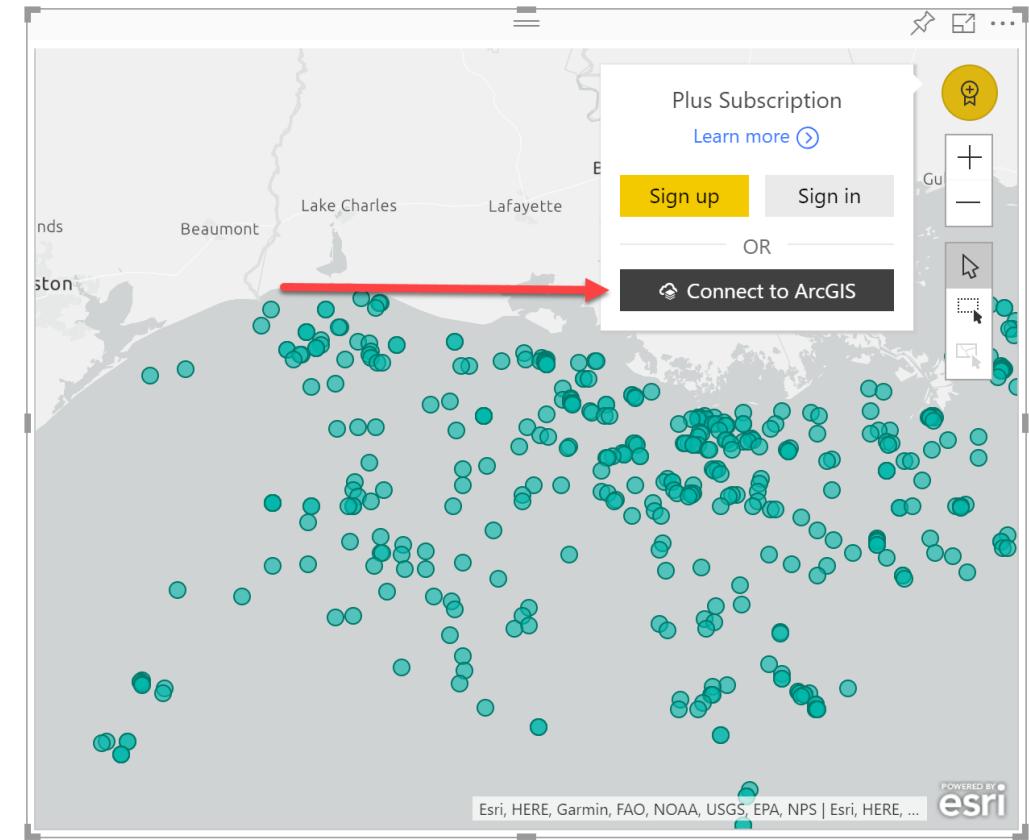
Map visuals – ArcGIS

**Sign up / Sing in to ESRI PLUS or connect to ArcGISOnline from the ArcGIS Visualization**

**Plus subscription** provides additional base map and global demographics

**ArcGIS Online** provides secure organizational data for your reports

Can put up to **5,000 geo locations** on your map



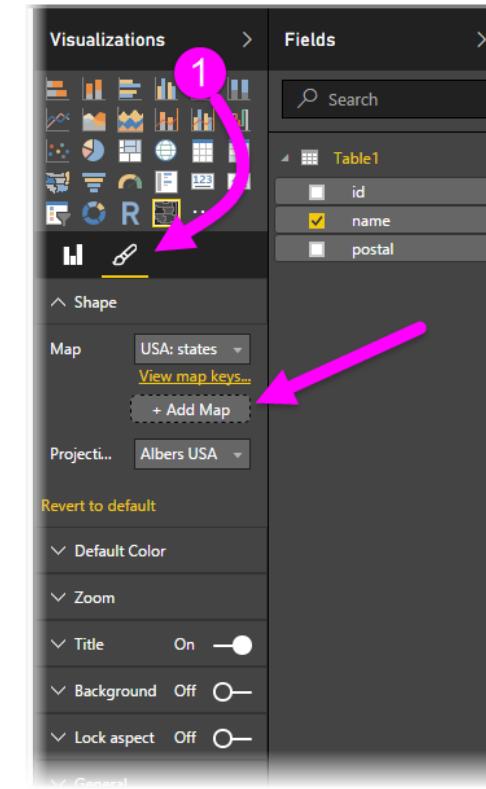
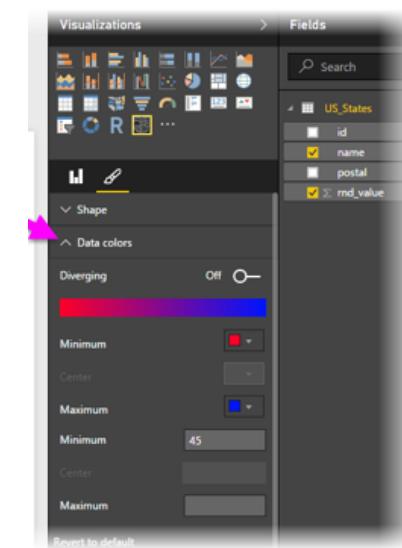
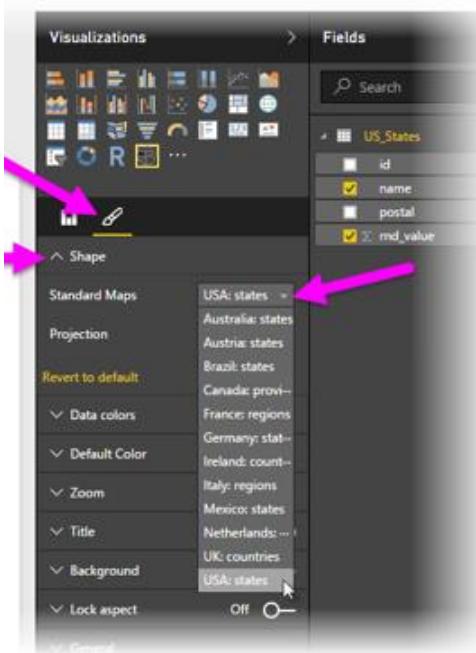
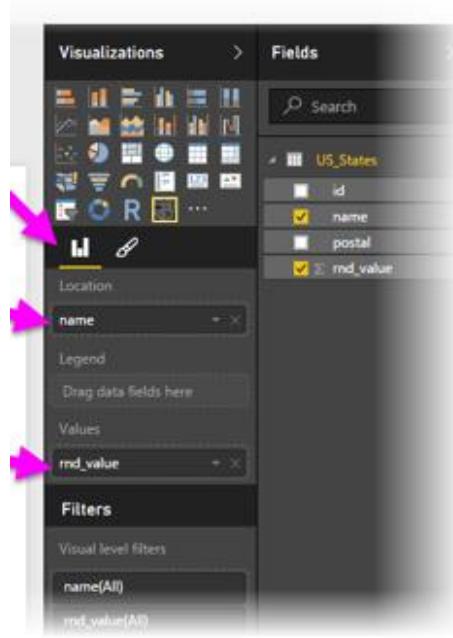
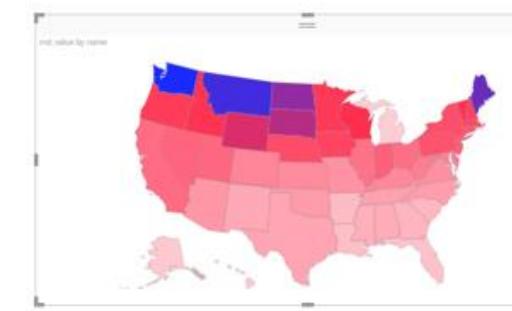
# Building a Report

## Map visuals – Shape map (Preview)

Shows relative comparisons of regions on a map by coloring them differently

Based on **ESRI/TopoJSON** maps

Supports **custom maps**

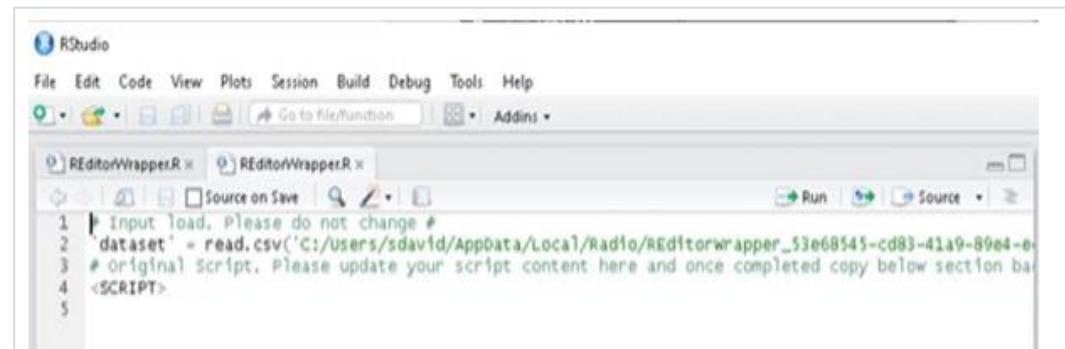
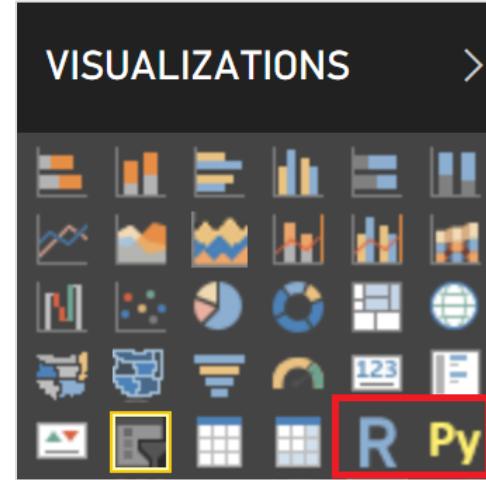
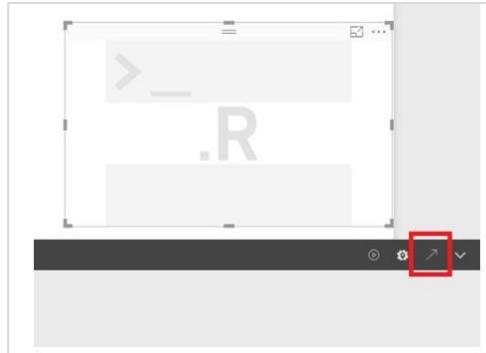
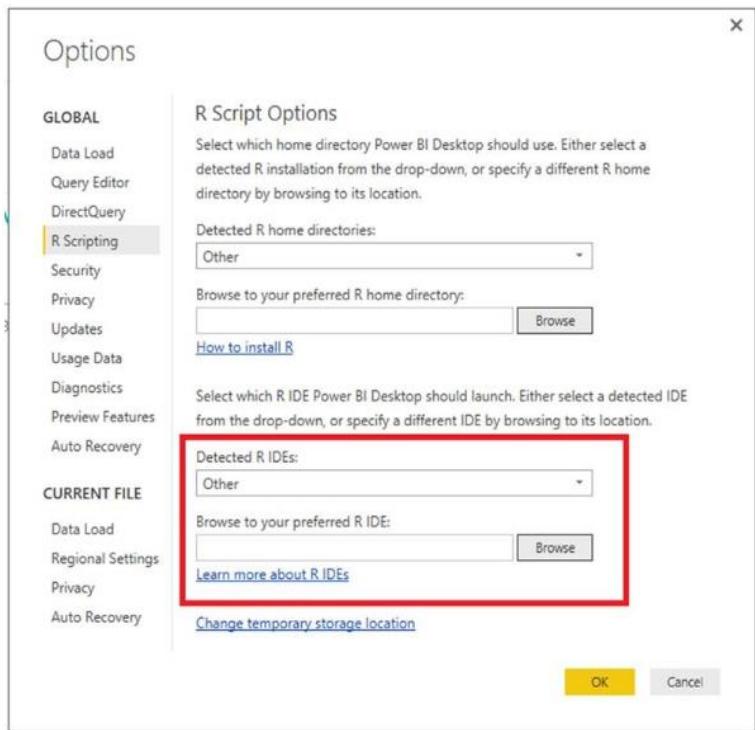


# Building a Report

## Visualizations – ML Script visuals

Create charts  
based on **R or**  
**Python scripts**

Choose your  
favorite **IDE** for  
both, to edit script

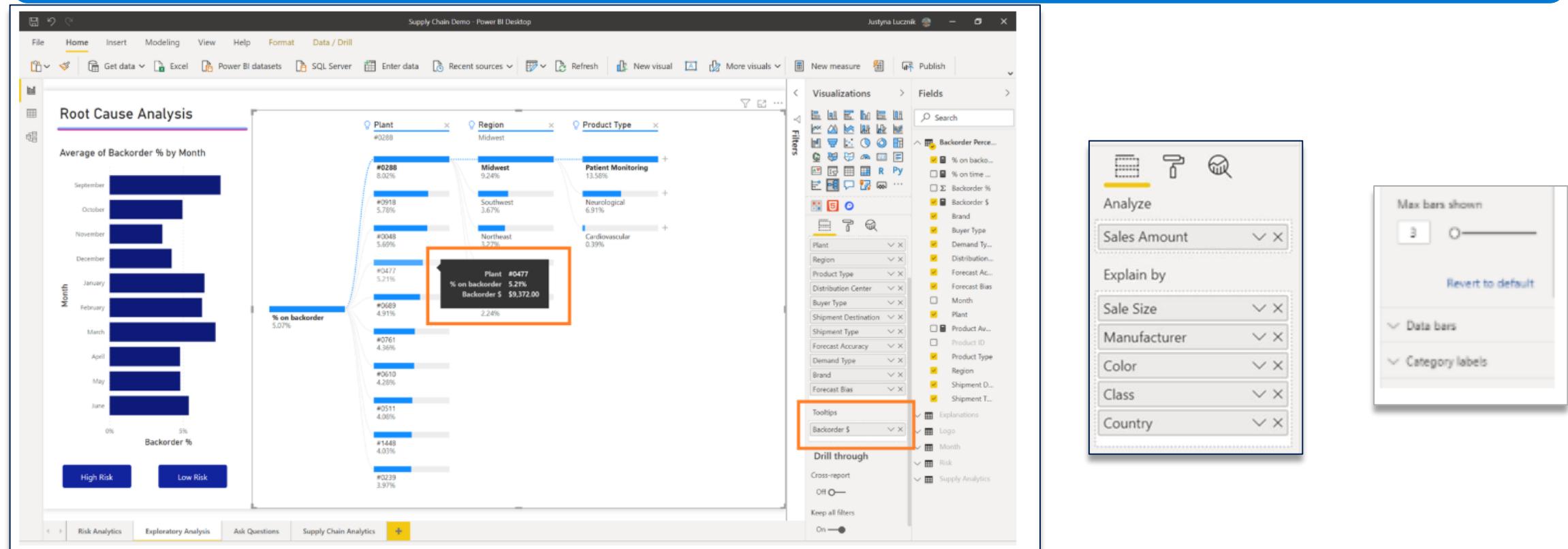


# Building a Report

## Advanced visualizations – Decomposition Tree Visual

Used to perform **root-cause analysis** by viewing how individual categories in a group contribute to the whole.

The visual lets you **decompose**, or break down, a group to see its individual categories and how they can be ranked according to a selected measure, such as by sales amount.



# Building a Report

## Advanced visualizations - Key Influencer Visual

Understand the **factors** that drive a metric you're interested in by analyzing your data, ranking the factors that matter, and displaying them as **key influencers**.

For example, suppose you want to figure out what influences' employee **churn**. One factor might be **employment contract length**, and another one might be employee **age**.



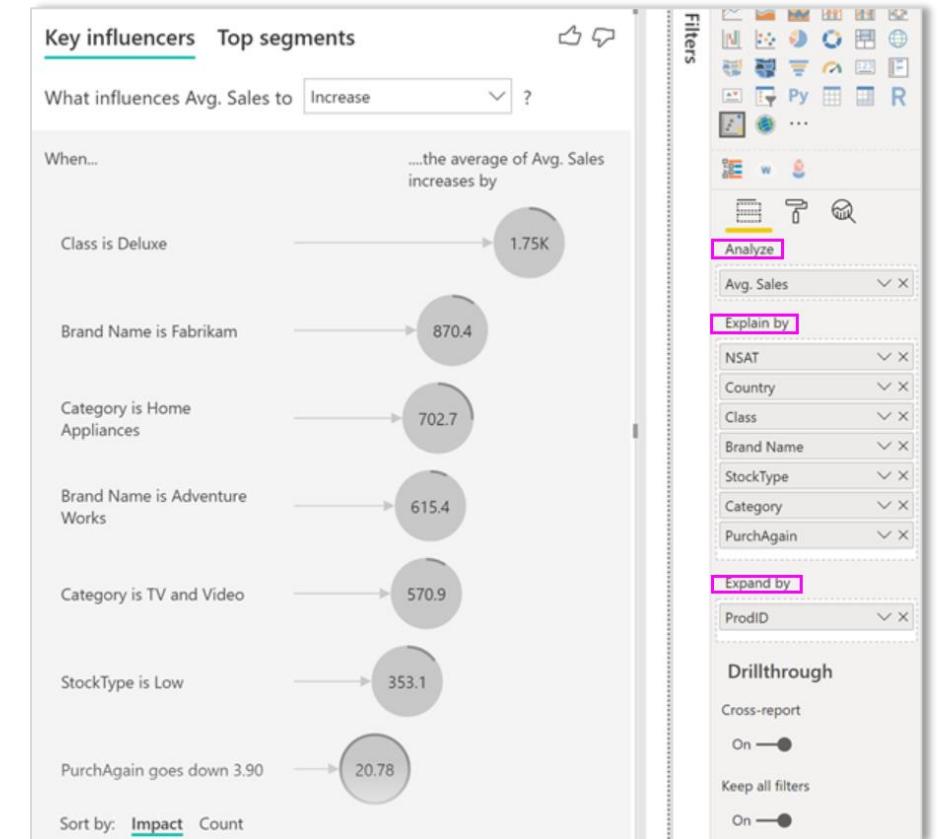
# Building a Report

Advanced visualizations - Key Influencer Visual

In the Analyze bucket, **numerical**, and **categorical** fields, with Aug 2019 update **measures** can be used.

By **default**, measure is evaluated based on the fields in the **Explain by** bucket.

If you want it to be evaluated over all your data, add an ID field to the **Expand by** bucket.



# Building a Report

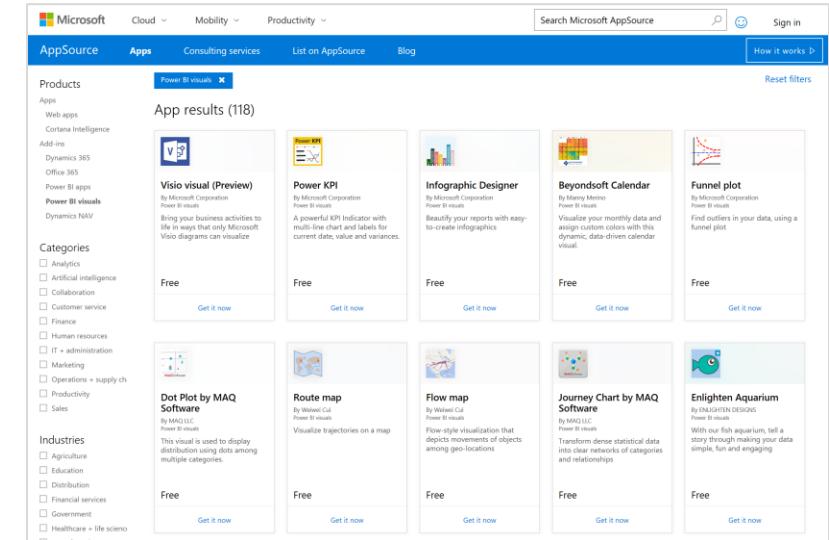
## Visualizations – Custom Visualizations

Ready-made custom

You can also **build** your own visualizations (step by step **tutorials** available)

After you build them, **you can use them** or **you can share with the community**

If you share, we will do a quality check and make it **available at our marketplace**



# Demonstration

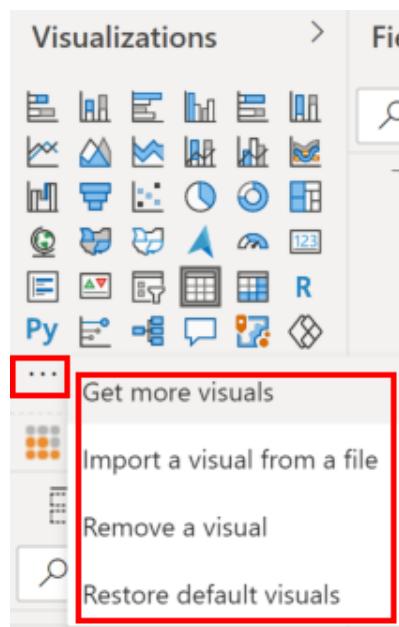
## Conditional formatting



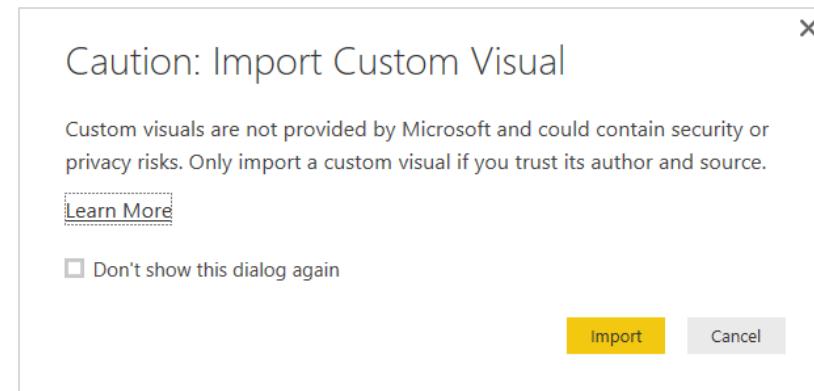
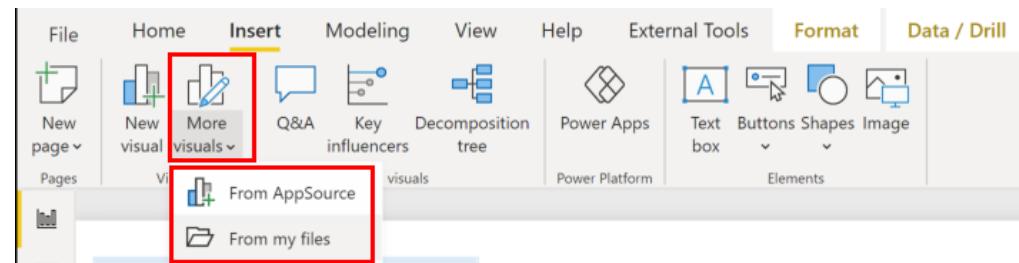
# Building a Report

## Visualizations – Custom Visualizations

- Download the visual from store (AppSource) or from file (.pbviz file extension)



OR

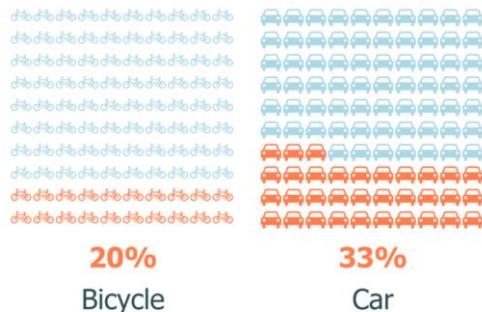


# Building a Report

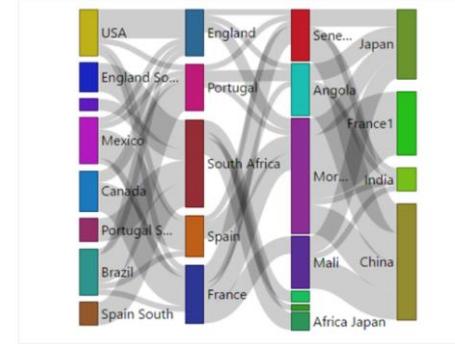
## Visualizations – Custom Visualizations (examples)



Synoptic Panel by OKViz



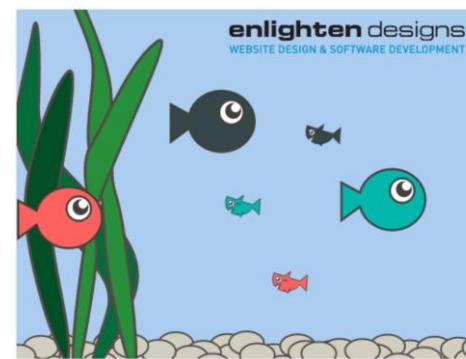
Waffle Chart



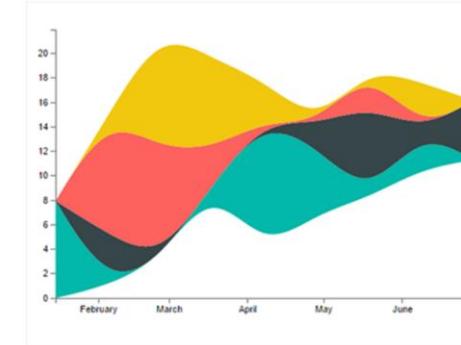
Sankey Chart



Word Cloud



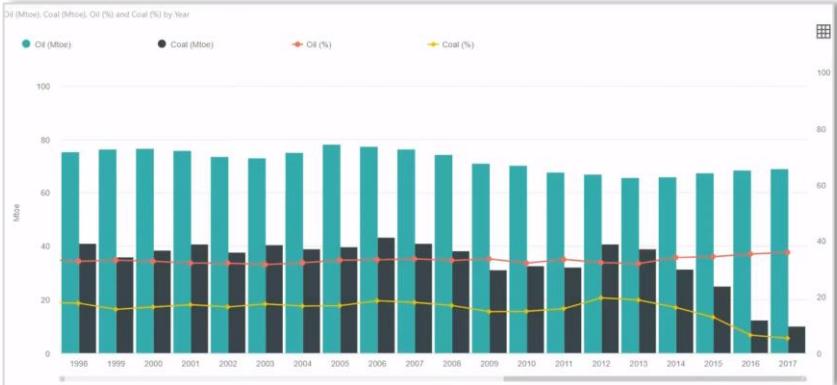
Enlighten Aquarium



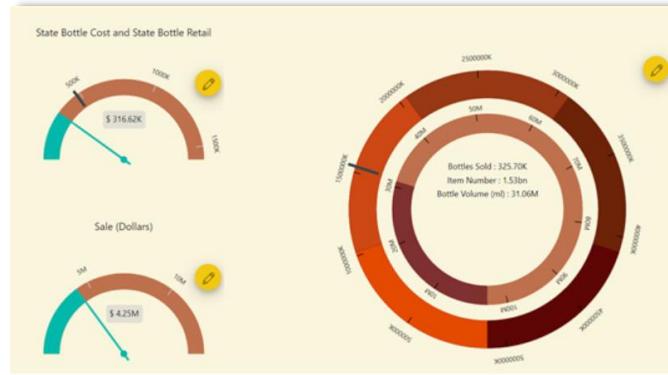
Stream Graph

# Building a Report

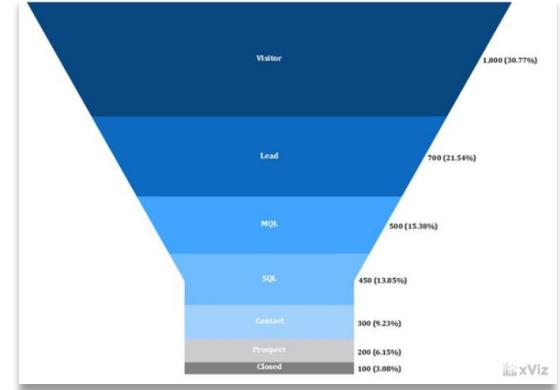
## Visualizations – Custom Visualizations - xViz Suite (examples)



Multiple Axis Chart



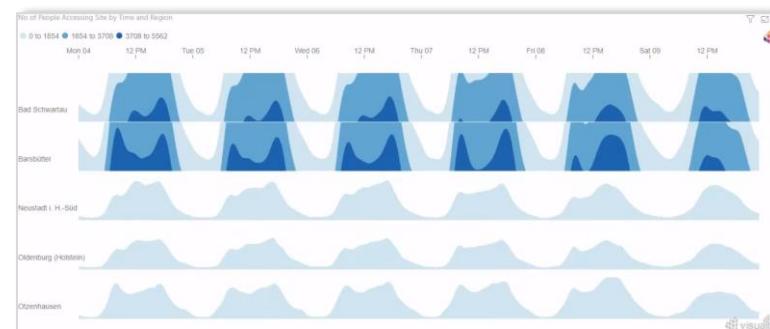
Advanced Gauge Chart



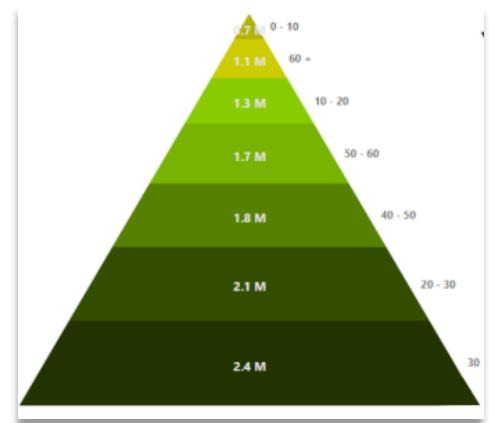
Funnel Chart



Variance Chart



Horizon Chart



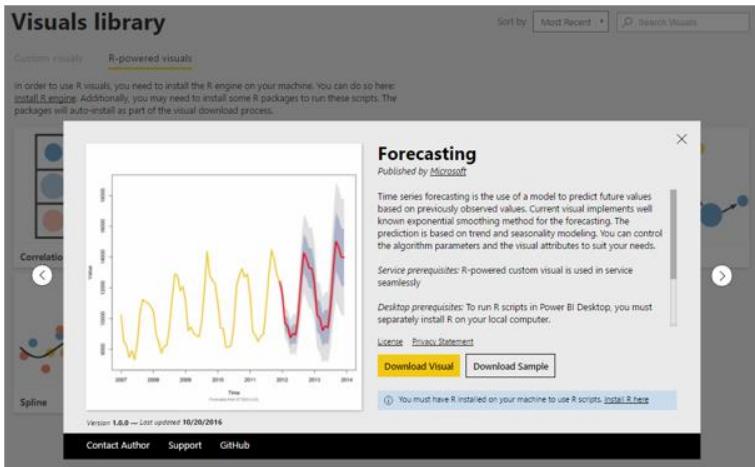
Pyramid Charts

# Building a Report

## Visualizations – Custom Visualizations

R-powered custom visuals

**Download and import** the R-powered custom visual like any regular custom visual

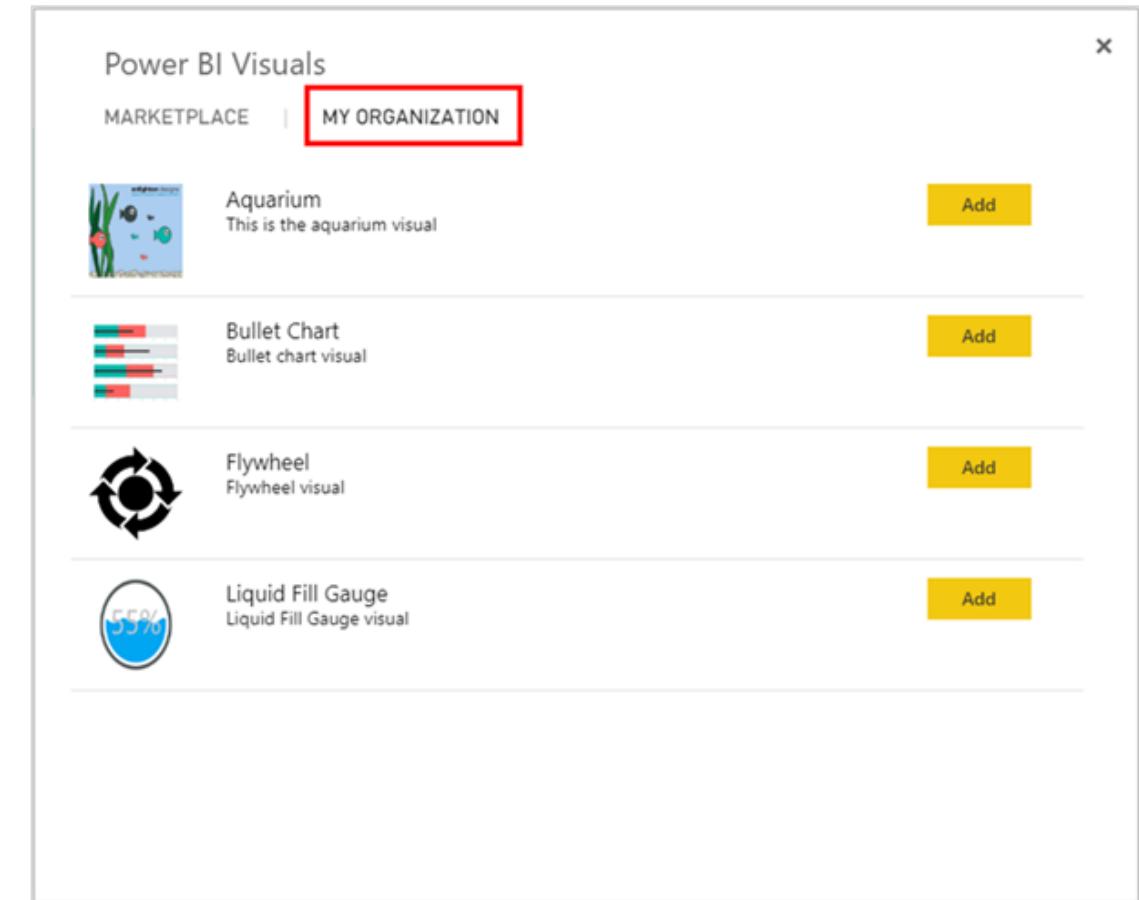


A screenshot of the Power BI desktop interface. The 'Visualizations' pane on the left shows various chart and map icons. The 'Fields' pane on the right lists fields such as Date, Value, and Price, each with checkboxes for 'Date', 'High', 'Low', 'Open', and 'Price'. The main workspace shows a chart with a blue line and red dots.

# Building a Report

Visualizations – Organizational Custom Visuals

Power BI admin can **deploy specific  
visuals as part of your  
organizational repository**



# Building a Report

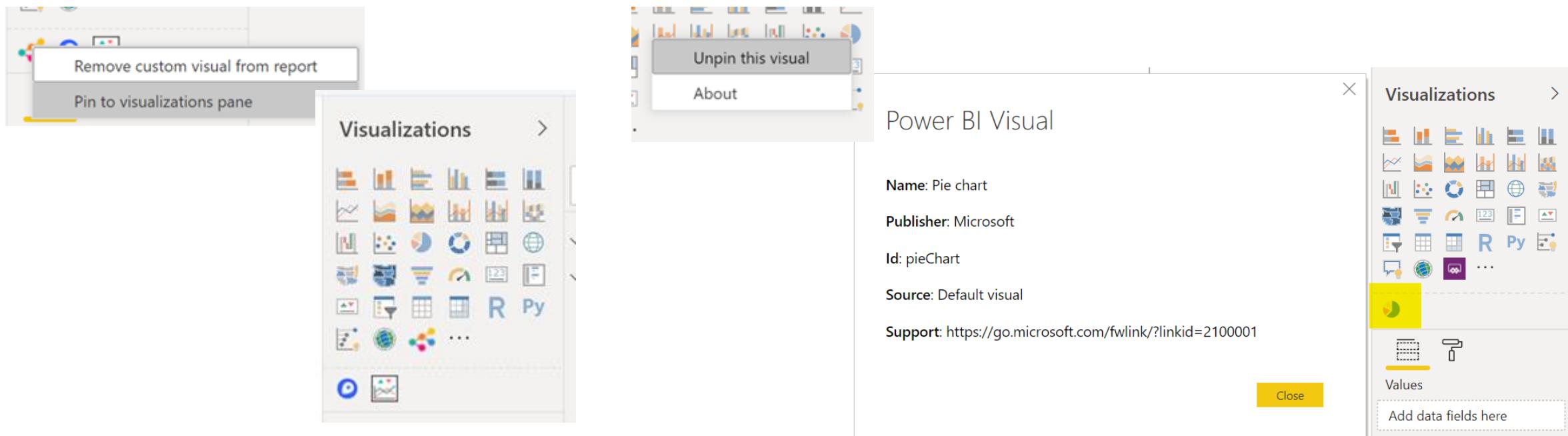
## Personalized visualization pane

Pin any marketplace visual to your pane and have it show up automatically for you every time you open any report

Makes it very easy to standardize on a visual **without needing to add it to every single report**

Unpin default visuals you do not wish to see.

About shows the visual's Name, Publisher, etc.



# Demonstration

## Custom Visuals



# Building a Report

Interactive Features - Drill down and up

---

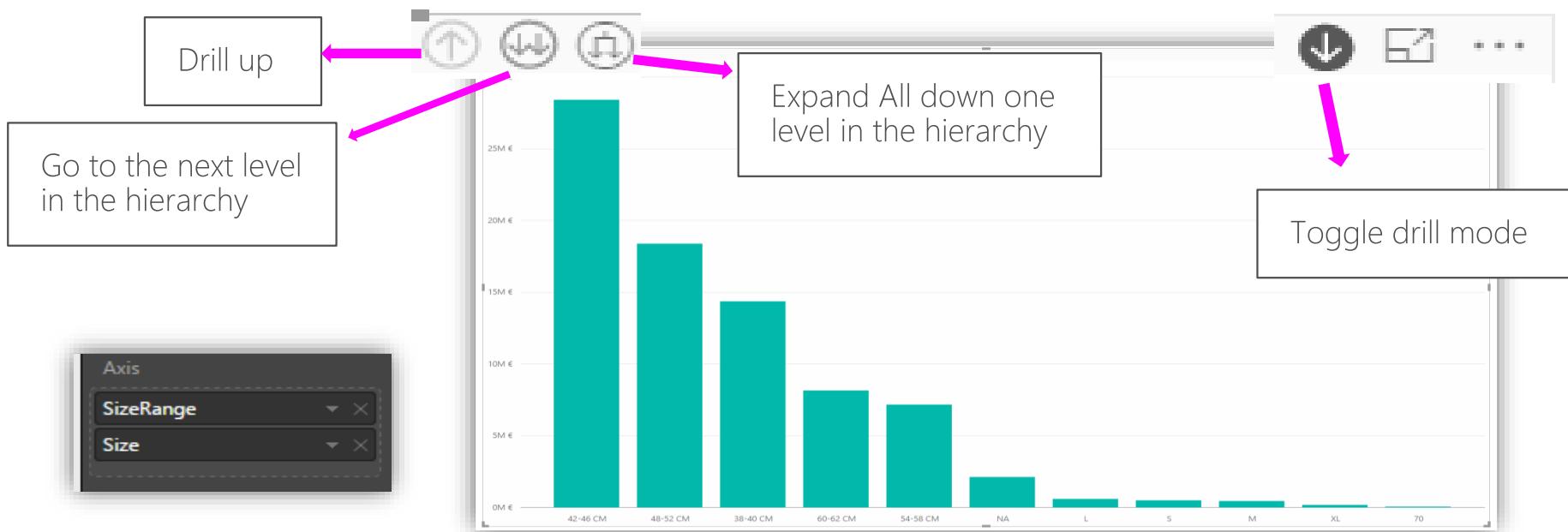
It is possible to **navigate across attributes** if they are stacked on a particular area of a visualization (down/up/across)

---

Those attributes don't have to be **natural hierarchies**, all of them can be mixed.

---

**Drill features works on bar and line charts**

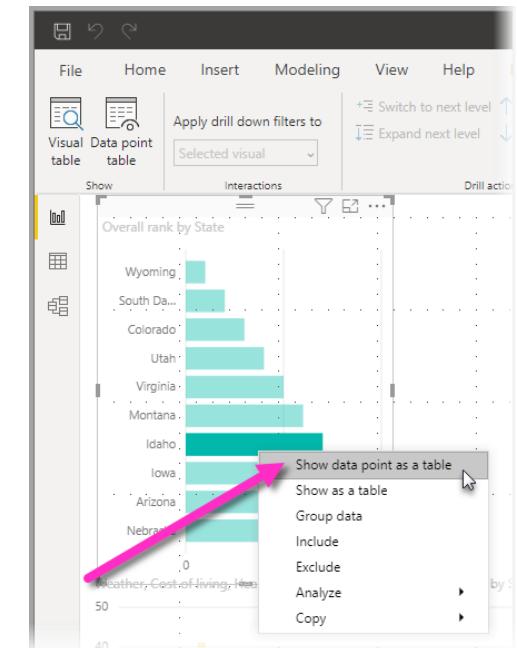
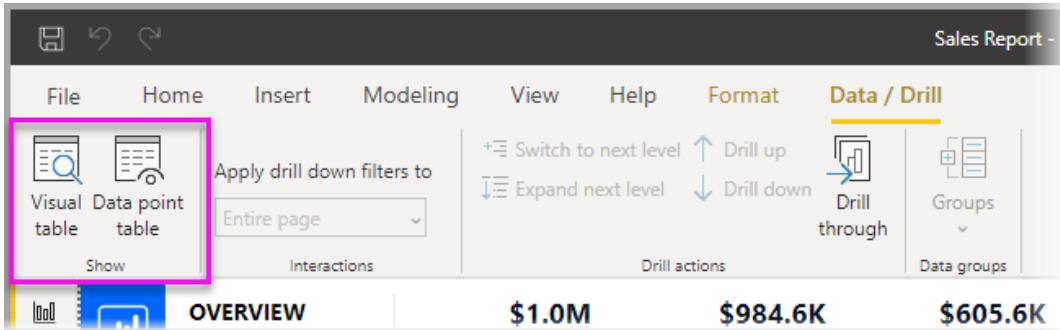
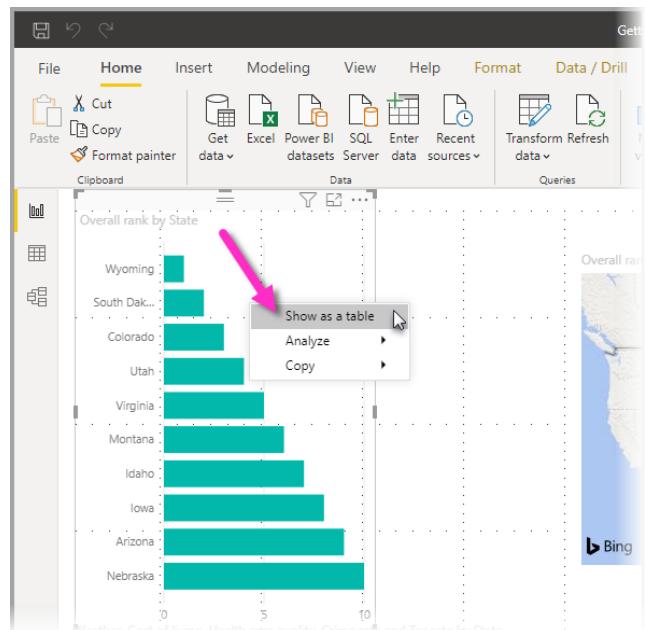


# Building a Report

## Interactive Features - Drill to records

Drill **ribbon** allows controlling of **drilling operations**

Use **Visual table** to view the data in a visual as a table, or use **Data point table** to view a table of the data used to calculate a single data point

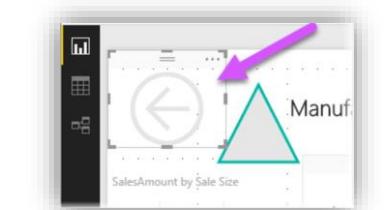


# Building a Report

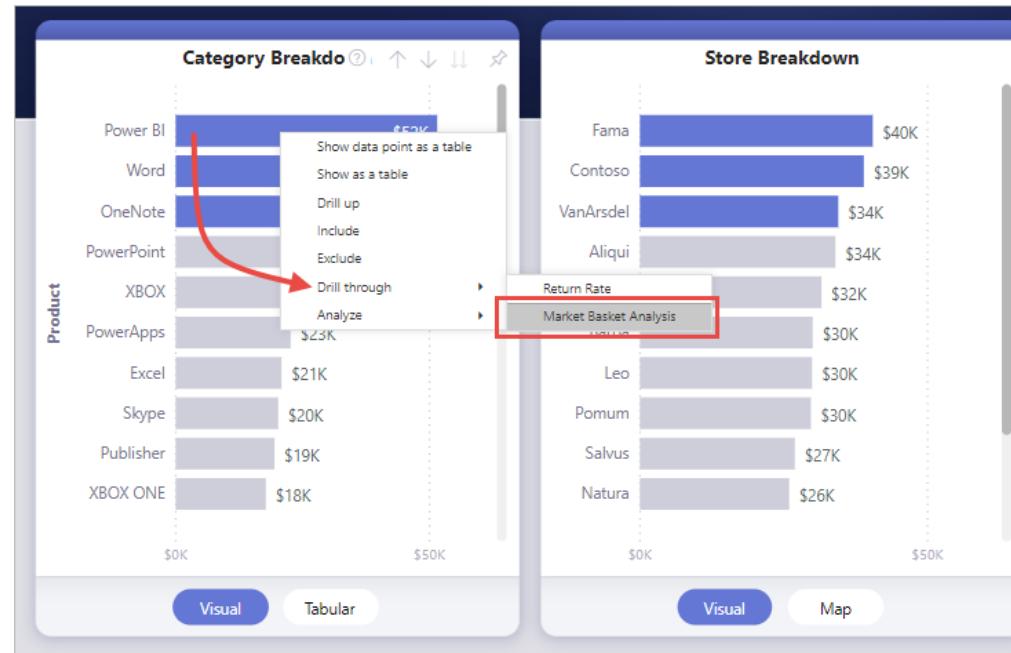
## Interactive Features - Drill through

It is possible to create a page in your report that **focuses on a specific entity** - such as a supplier, or customer, or manufacturer.

With that focused report page, users can right-click on a data point in other report pages, and drill through to the focused page to get details that are filtered to that context.



Back button created automatically



Drill through option on the visual

Visualizations > Fields >

Search

- Calendar
- Channel
- Geography
- Product
- ProductCategory
- ProductSubcategory
- Promotion
- Sales
- Stores

Values

Add data fields here

Drillthrough

Cross-report

Off

Keep all filters

On

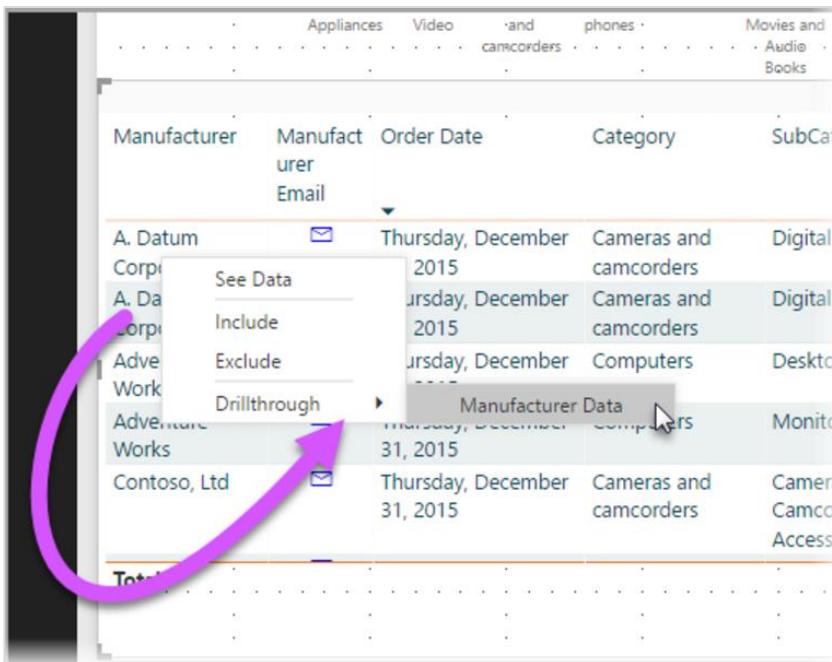
Add drillthrough fields here

Adding the drill through fields

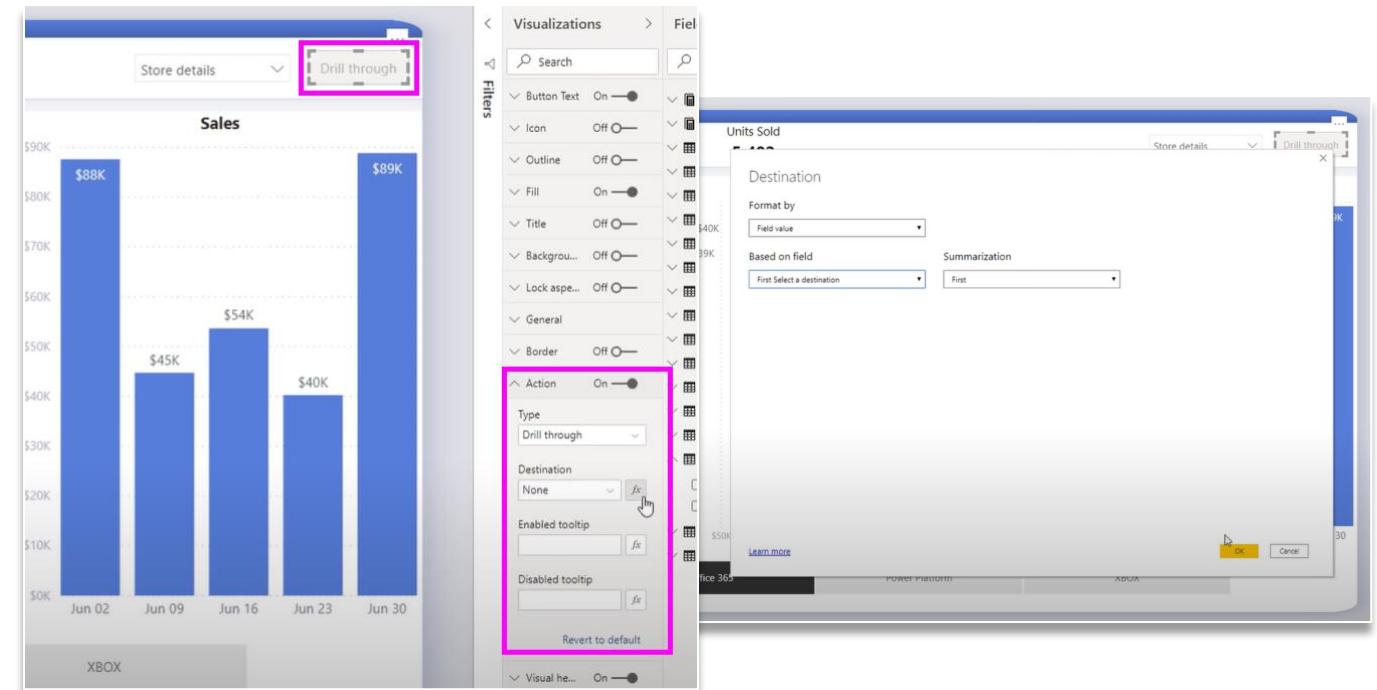
# Building a Report

## Interactive Features - Drill through

- Right-click on a **data point** configured for drill through
- A **context menu** appears, letting you drill through to that page.



- Conditionally set the drill through **destination** for the button
- Conditionally format the **tooltips** for the enabled and disabled button state
- Individually **customize** the formatting for the disabled state of the button



# Building a Report

## Interactive Features - Drill through (Measures)

Measures can be passed as a drillthrough filter

Can **pass all filters** and not just specific fields placed in the bucket by turning on Pass all filters

All filter context from the source page is passed to the drillthrough page, which is now shown on the Drillthrough bucket.

### Drillthrough

Cross-report

Off

Keep all filters

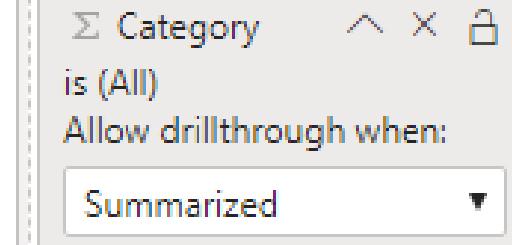
On

Σ Category ^ X 

is (All) ▼

Allow drillthrough when:

Summarized



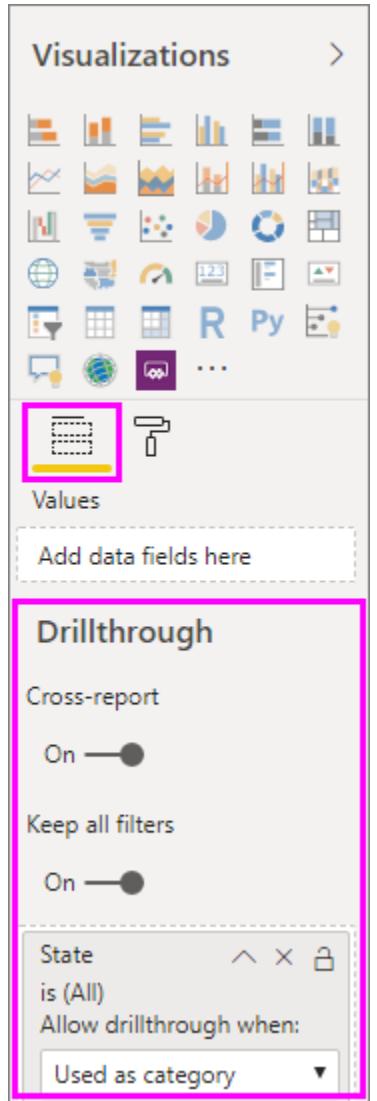
# Building a Report

## Interactive Features - Drill through (cross-report)

Can be used to refer other reports in the same workspace. Through this you can setup links between reports in the same workspace.

### Steps to enable

- Set up a drillthrough target page to be accessed from other reports within a workspace.
- Allow a report to opt into seeing drillthrough pages outside of the report.
- In the Report settings for the current file section of the Options dialog, enable the Cross-report drillthrough setting for all reports within a workspace that you want to point that cross-report drillthrough page



# Building a Report

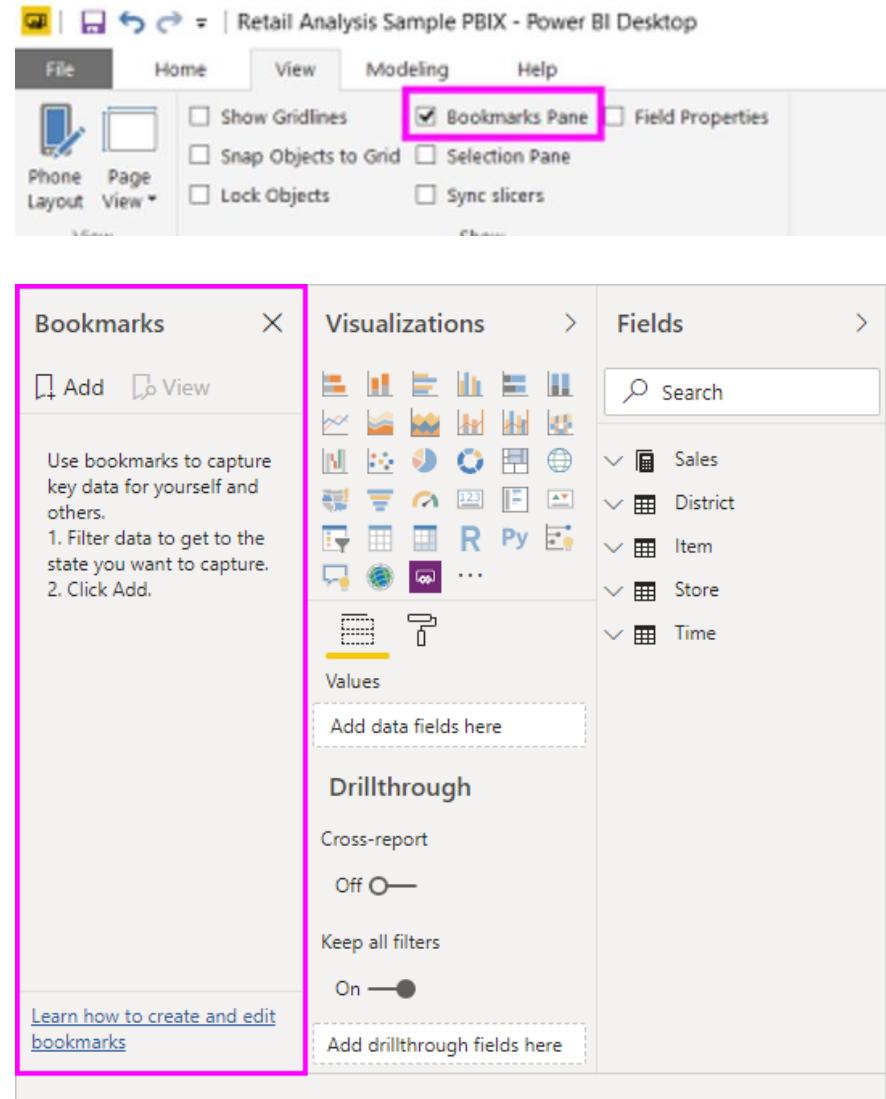
## Interactive Features - Bookmarking

**Save interesting states** as part of your report.

Pick up where you left off in your analysis.

Create **PowerPoint-like presentation** of your report.

Create **report navigation**, and more.



# Building a Report

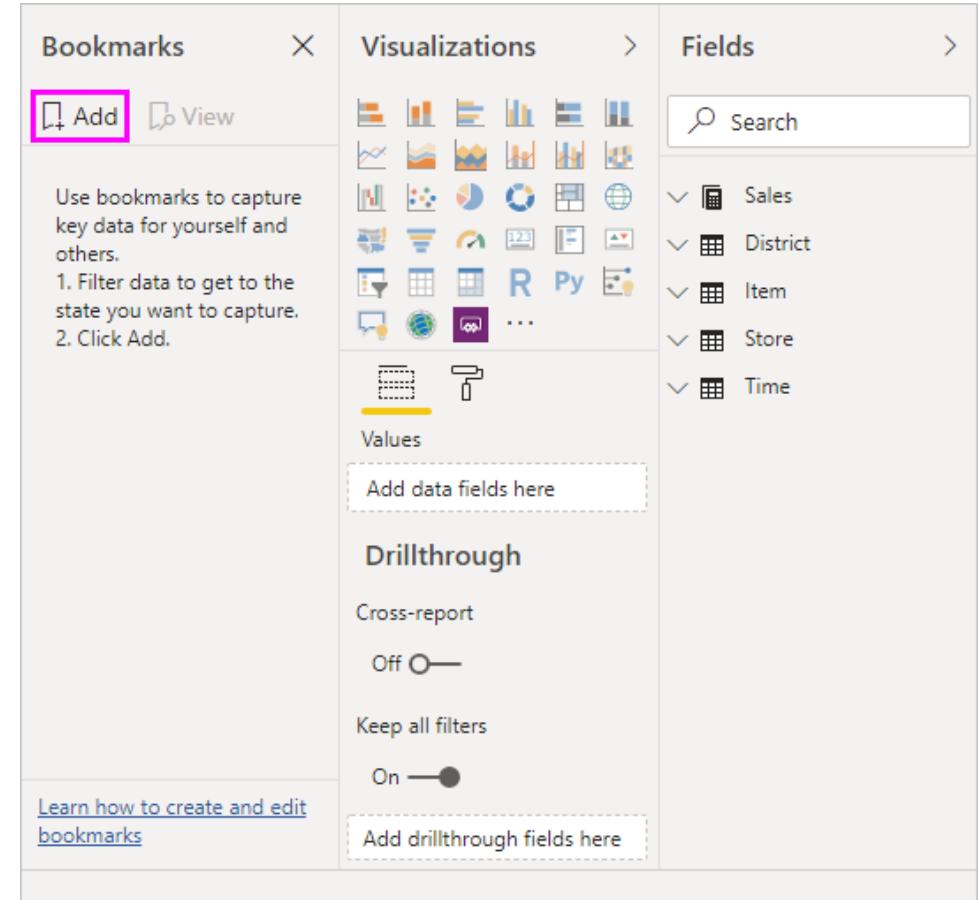
## Interactive Features - Bookmarking

**Set up report** to look how you want and **click the Add button** under Bookmarks.

**Rename or rearrange** the bookmarks, as necessary.

Use bookmark by **simply clicking a bookmark** from the bookmark pane; or

Use **View option** to use bookmarks **as a story or in a presentation.**



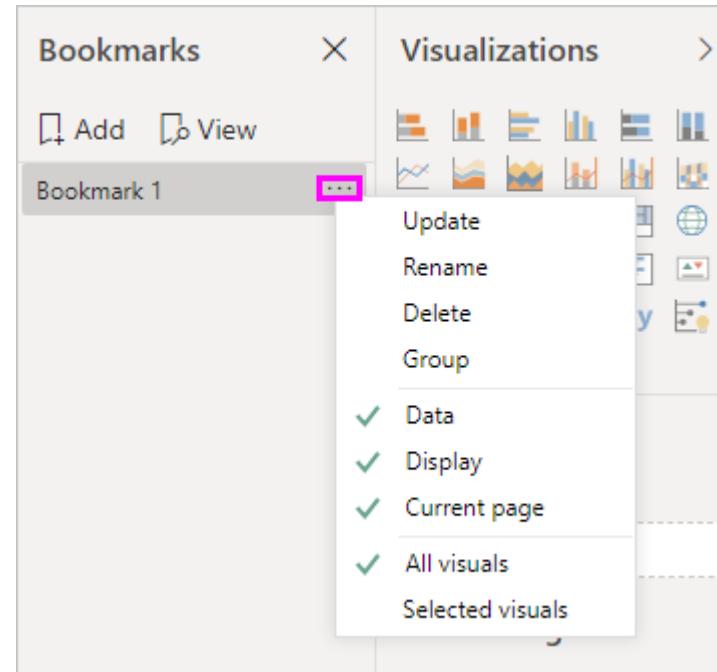
# Building a Report

## Interactive Features - Bookmarking

Bookmark will include the following:

- the current page
- Filters
- Slicers
- Sort order
- Drill location
- Visibility (more on this later)
- Any of the “focus” modes

**Bookmark links are available for shapes and images; great feature for creating custom navigation in reports.**



The screenshot shows the 'Format shape' pane in Power BI. The 'Action' section is set to 'On'. The 'Type' dropdown is set to 'Bookmark'. A dropdown menu lists several bookmark options: None, Overview, Top ten rankings, Weather comparison, Cost of living comparison, Ten most challenging, and Summary findings. The 'Bookmark' option is highlighted with a pink box.

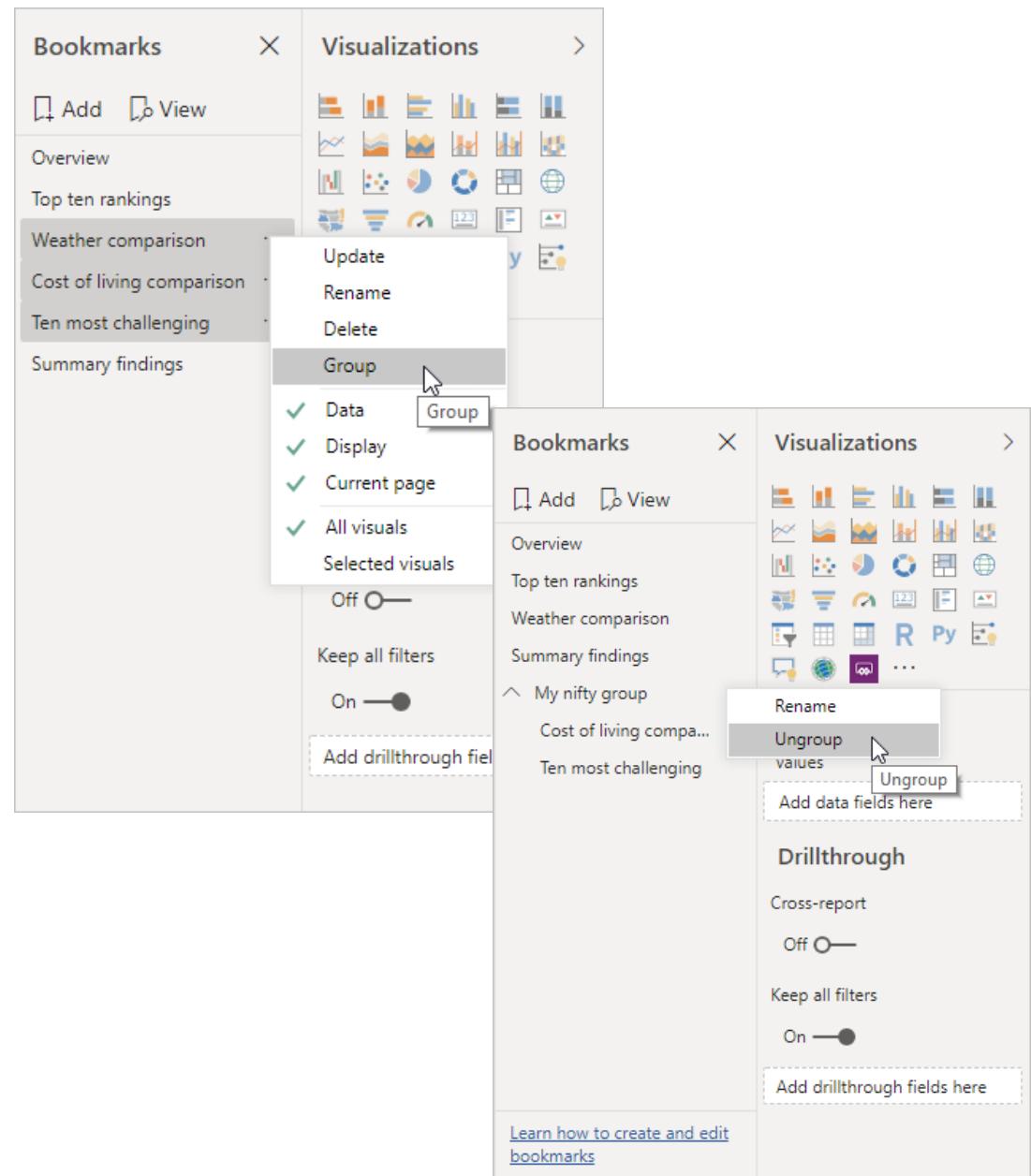
# Building a Report

## Interactive Features - Bookmark groups

Create **sets of bookmarks** with different purposes.

Collect a set of **bookmarks together** and give it a **name**

Each group can be **expanded and collapsed for easy viewing**



# Building a Report

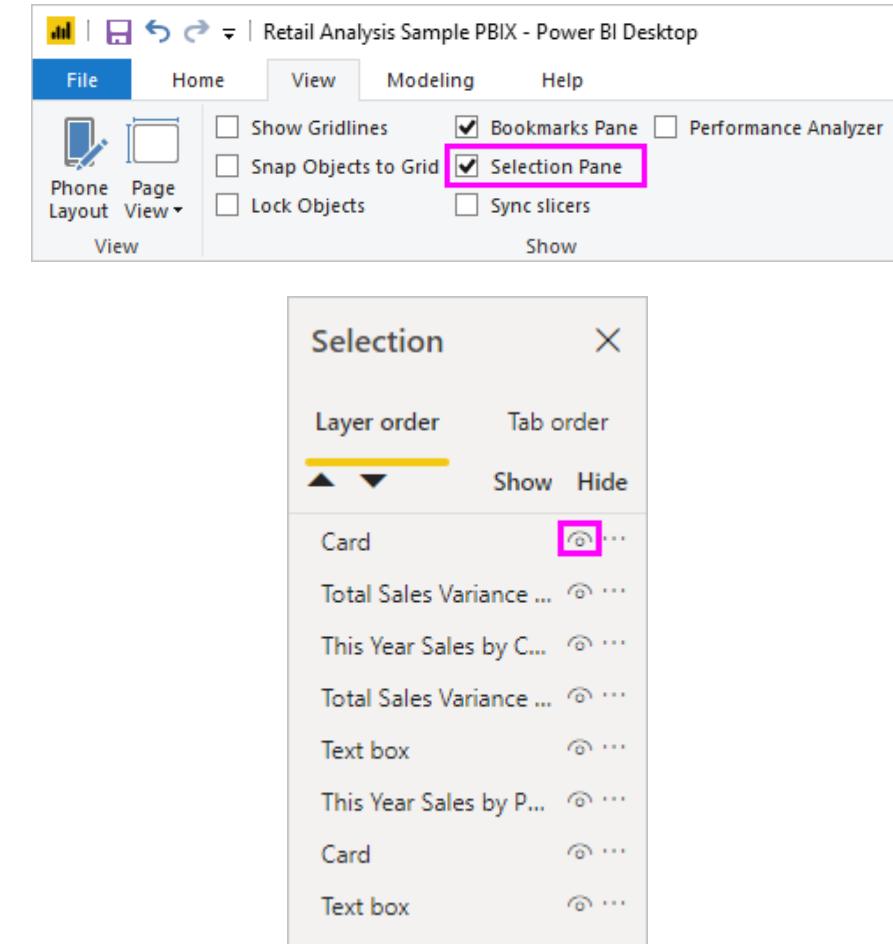
## Interactive Features - Selection Pane

**Lists out all objects** on the current report page and allows you to **select the visual** from here.

**Hide visuals** on the report through this pane.

**Combine this feature with bookmarking** to create interesting reports.

Very useful feature for **storytelling** purposes.



# Building a Report

Interactive Features – Selection Pane (Control order)

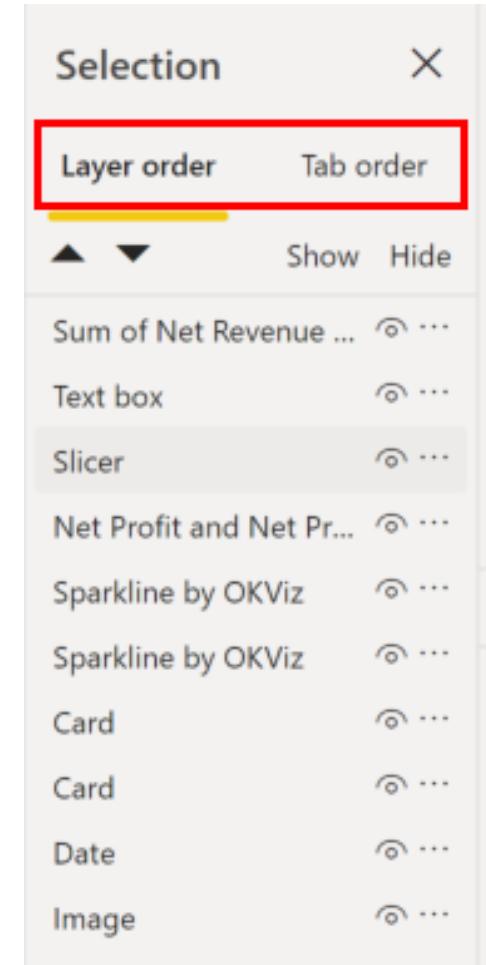
---

Easily change which **overlapping objects** show on **top on the canvas**

---

Objects **higher in Selection list** show **in front of** objects lower in the list

---



# Building a Report

## Interactive Features – Selection Pane (Grouping Visuals)

Group visuals, shapes, textboxes, images, and buttons together just like in **PowerPoint**.

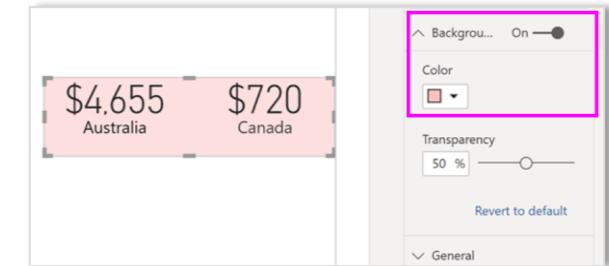
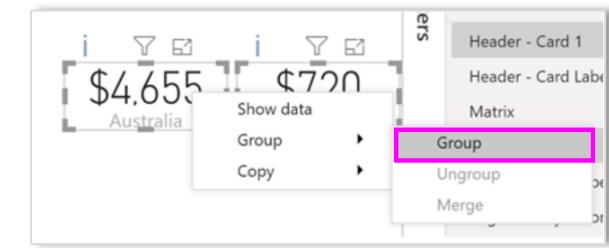
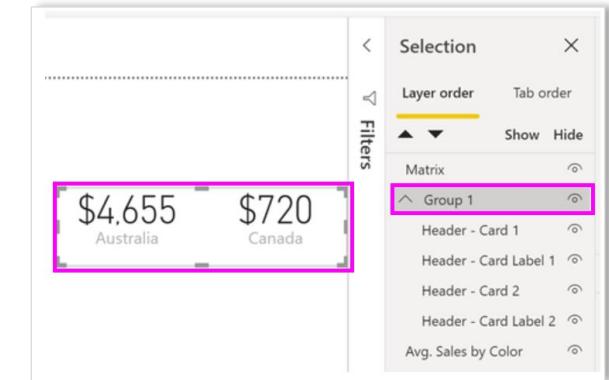
Control **visibility** of individually or as a group from the **selection pane**

Possible to **ungroup**

Possible to merge the groups into one group **instead of nesting them**

Clicking on the empty space within the group will not select anything

If it has **background color**, you will be able to select the group

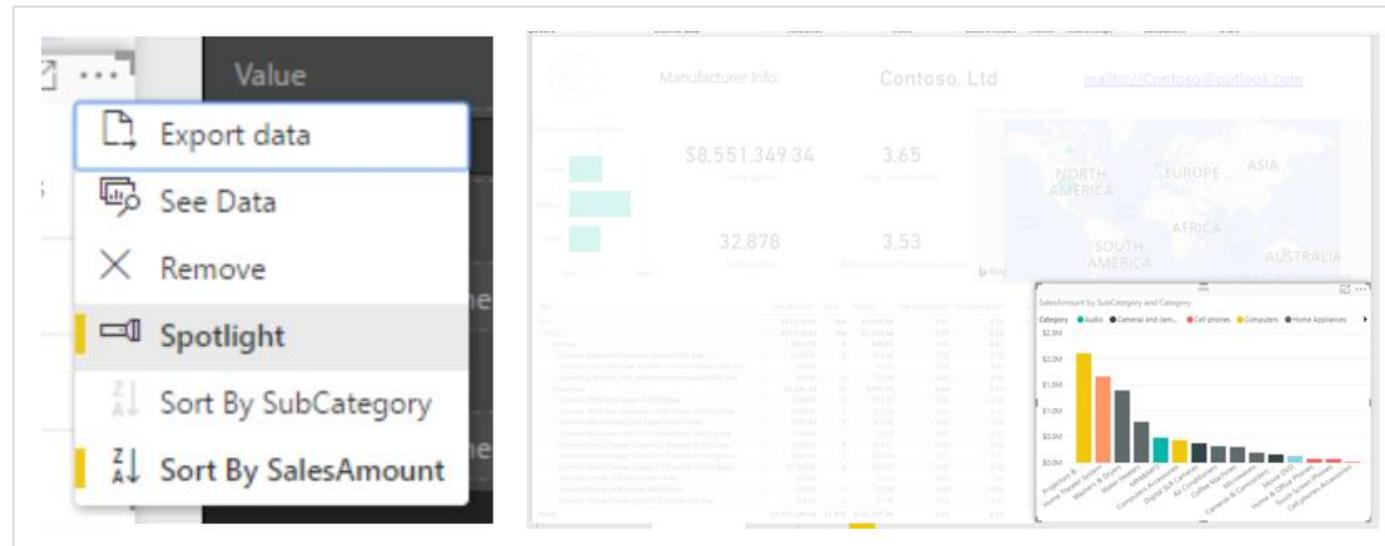


# Building a Report

## Interactive Features – Spotlight

**Call attention** to a specific visual **without losing context** during presentations.

This feature can be **included as part of a bookmark**.



# Building a Report

## Interactive Features – Buttons

Add interactive elements to **create app-like experience**

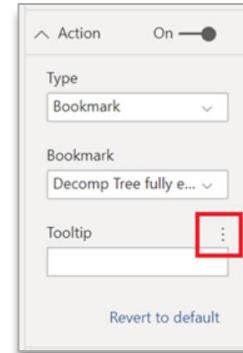
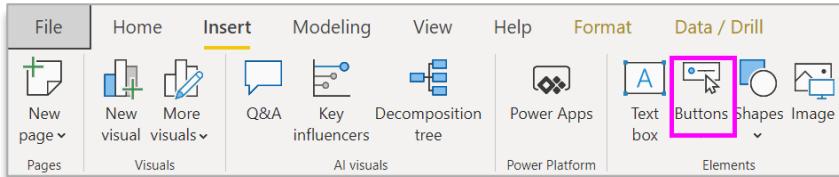
Several **button icons** including blank button available

**Assign actions** like 'back', bookmark, Q&A, page navigation and drill through to buttons

**Control format** of buttons including text, fill color, hover style, filling with images (for ex disabled, hover, click) and more

Also **conditionally formatting** of buttons & **conditionally setting** the drill through and page navigation **destination**

Use **Conditional Formatting** to customize the web URL actions.



This block contains several screenshots illustrating button configuration. 1. A 'Navigate to' dialog with a 'Market details' dropdown and a green 'Go' button. 2. A 'Action' settings dialog for a 'Page navigation' button, with the 'Destination' field set to 'None' and a red box around the 'fx' button. 3. A 'Destination' configuration panel with fields for 'Format by', 'Based on field', and 'Summarization'. 4. A 'Visualizations' pane showing a bar chart for 'Sales' with a red box around the 'Drill through' button. 5. A detailed 'Format' pane for a bar chart, with sections for 'General', 'Border', and 'Action'. The 'Action' section is expanded, showing 'Type: Drill through', 'Destination: None', and 'Enabled tooltip' and 'Disabled tooltip' fields, all with red boxes around them.

# Building a Report

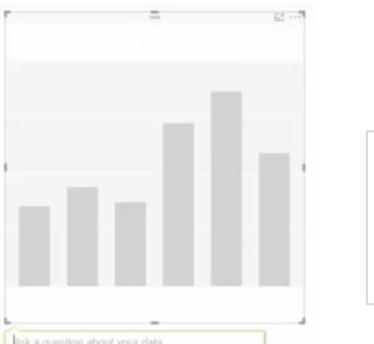
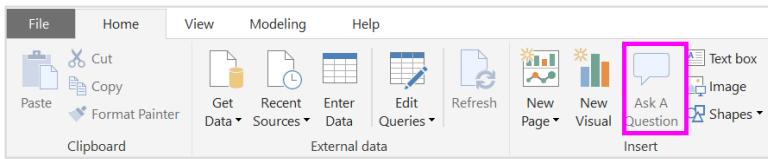
## Interactive Features – Q&A for Report Creation

Can set up an image, shape or button to **launch Q&A Explorer**

Report developers can add **suggested questions** for end-users

Q&A explored **auto-populates** with suggested questions.

With **Teach Q&A**, **definitions** and **measures** can be added to make Q&A work better with the data model.

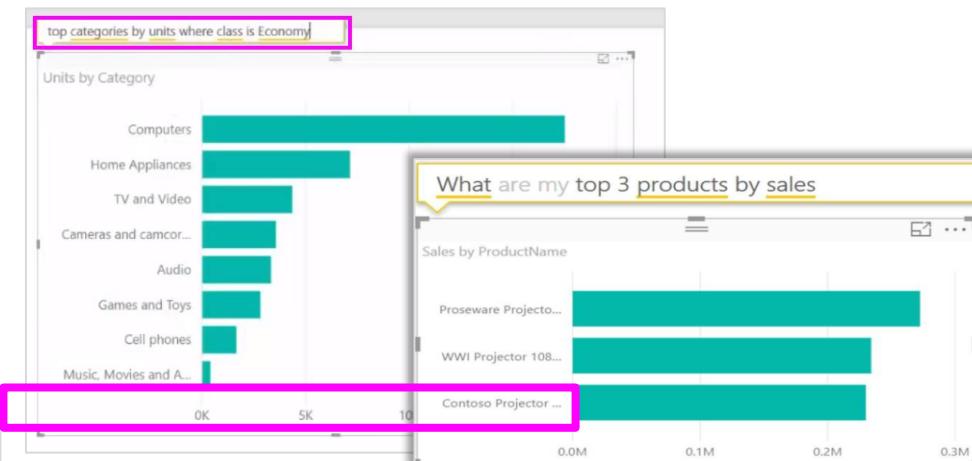


### Recommendations for Improving Results

products by m  
Did you mean: *products*

products by manufacturer  
Showing results for: *products by product manufacturer*

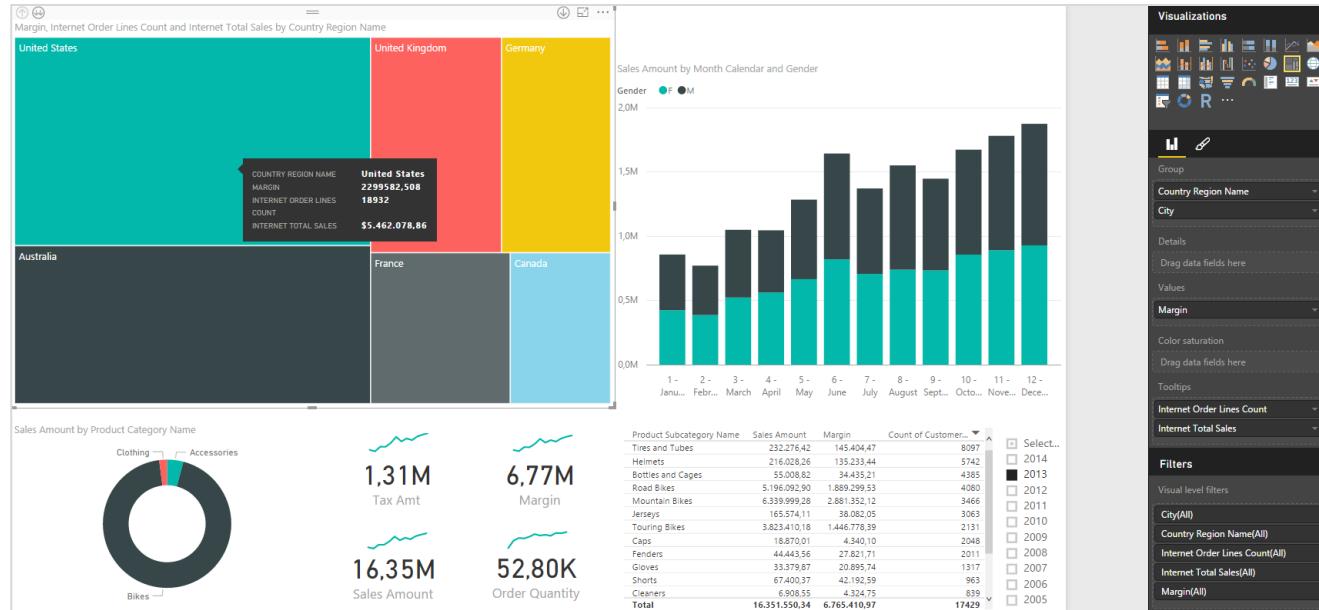
### Support Top/Bottom N selection



# Building a Report

## Interactive Features – Tooltips

- Provide **additional context when hovering** over a series in a visualization
- They are customizable but only Measures or aggregations can be shown



# Building a Report

## Interactive Features – Report Page Tooltips

Capability to **design a report page to be used as a custom tooltip** for other visuals in your report

**Full flexibility** to design whatever kind of custom tooltips you want, from using cards to show detailed information to miniature **drill through** pages **shown on hover**

### HOW TO USE:

- **Design the tooltip** you want to show
- **Mark the page as a tooltip** so we know you want to use it as such.
- To start using this tooltip in your report, **tie it to a specific field using the Tooltip fields bucket in the Filters pane**.



# Demonstration

**Drill through, Bookmarks and  
Storytelling example**

**Report Page Tooltips**

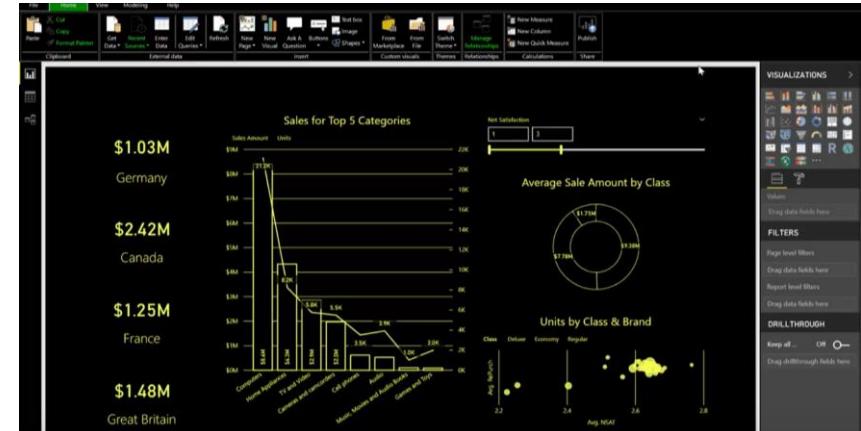
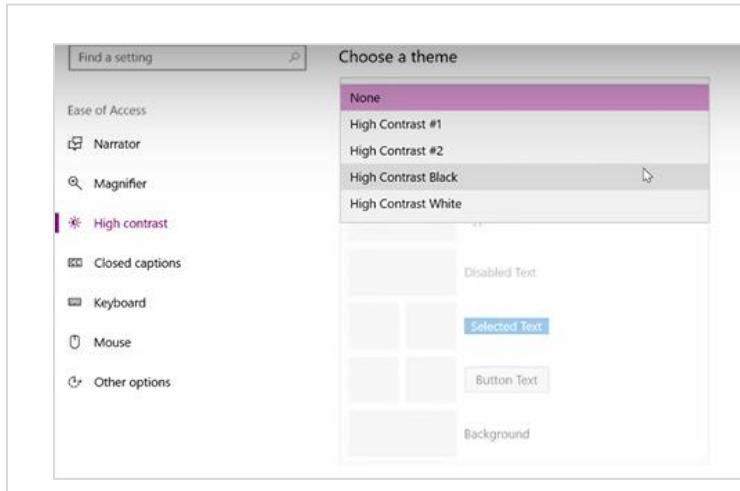


# Building a Report

More reporting features - High Contrast Support

High contrast mode Windows settings

When using Power BI Desktop, we'll **automatically detect** which high contrast theme you're using and apply those colors across your report, like the experience you're used to with other Microsoft products, such as Excel.



# Building a Report

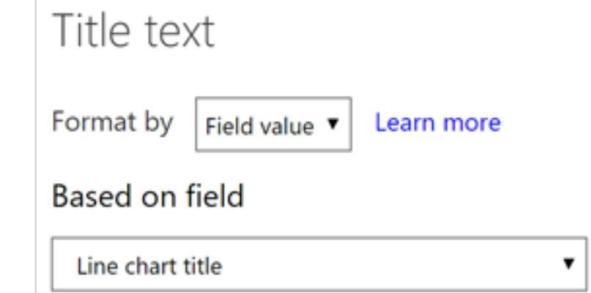
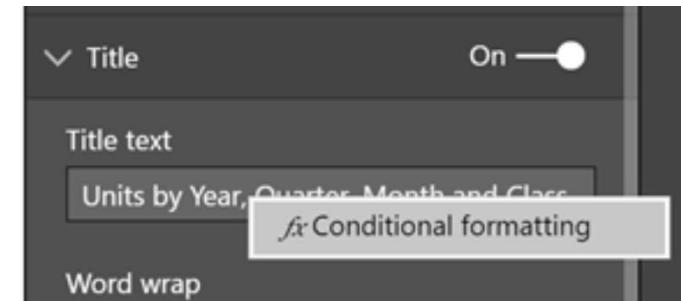
More reporting features - Conditional formatting for visual titles

Customizing the titles on the visuals is possible by using the conditional formatting.

Create a DAX Expression in the model to represent the title.

Launch the conditional formatting dialog by right-clicking the “Title text” area in the property pane card and picking Conditional formatting. In the dialog you can now pick the field you have created earlier. Same can be used to revert it.

```
1 Line chart title = "Units by Time and Class for " & SELECTEDVALUE(Sales[BrandName])
```



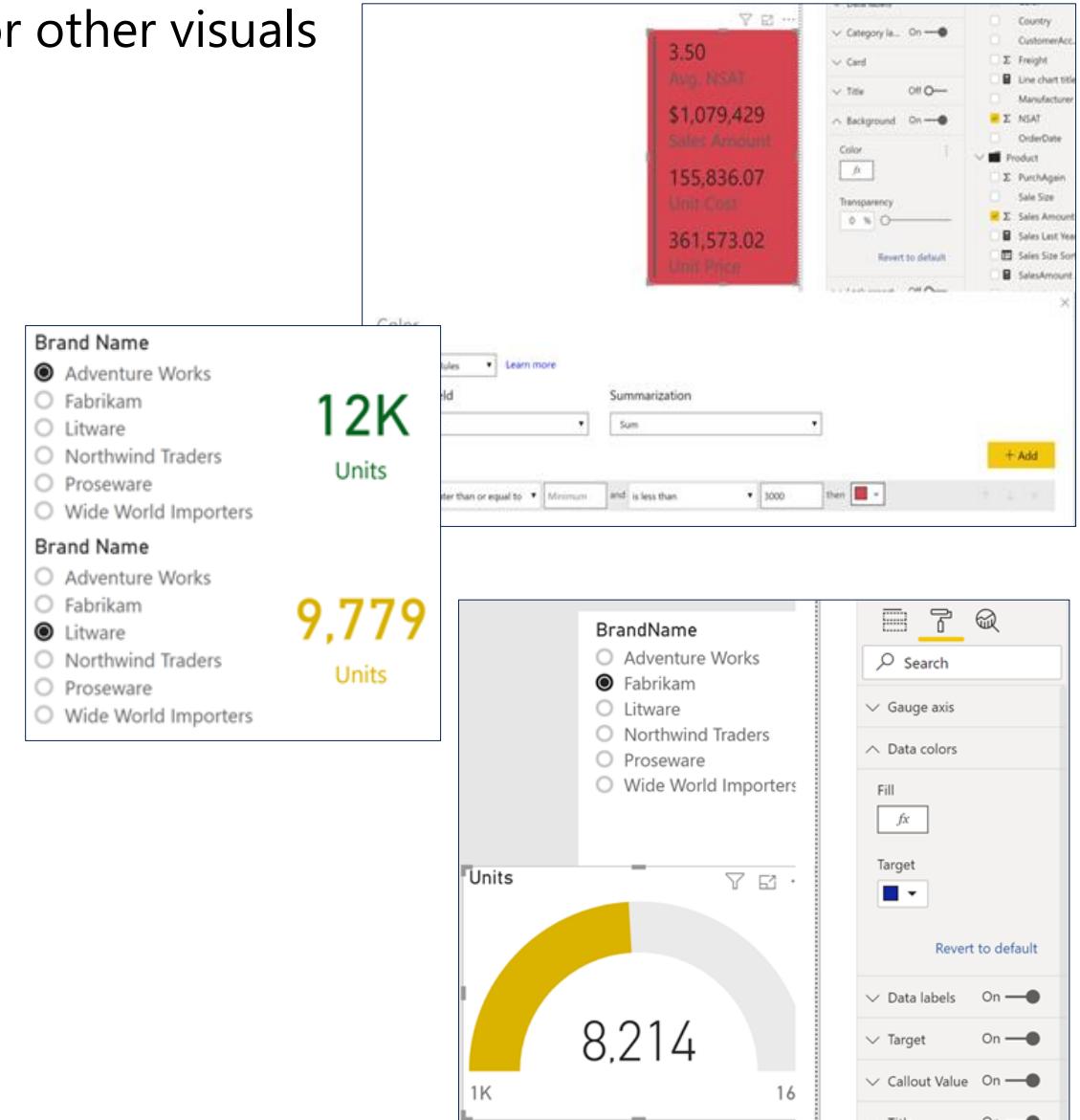
# Building a Report

More reporting features - Conditional formatting for other visuals

Any **visual's background** and both the visual title's **font color and background color** can be conditionally formatted

For **card visual**, you can now conditionally format the data label and category label colors

You can conditionally format the fill color of **gauge visual**



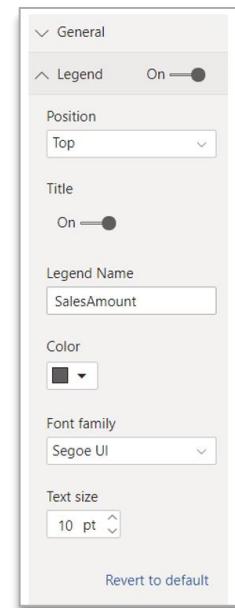
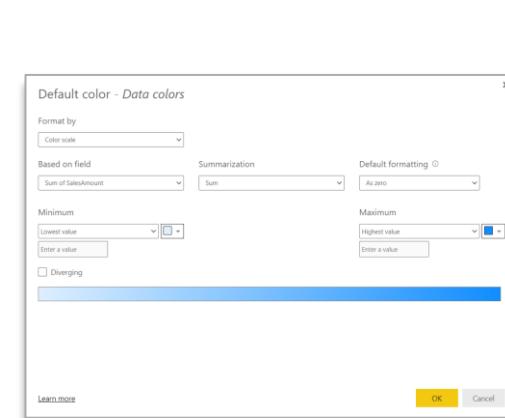
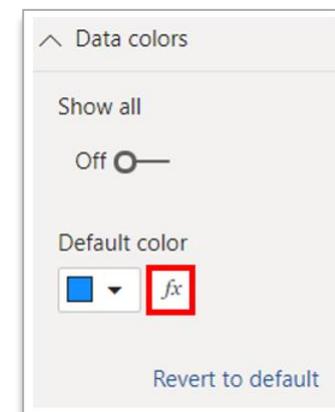
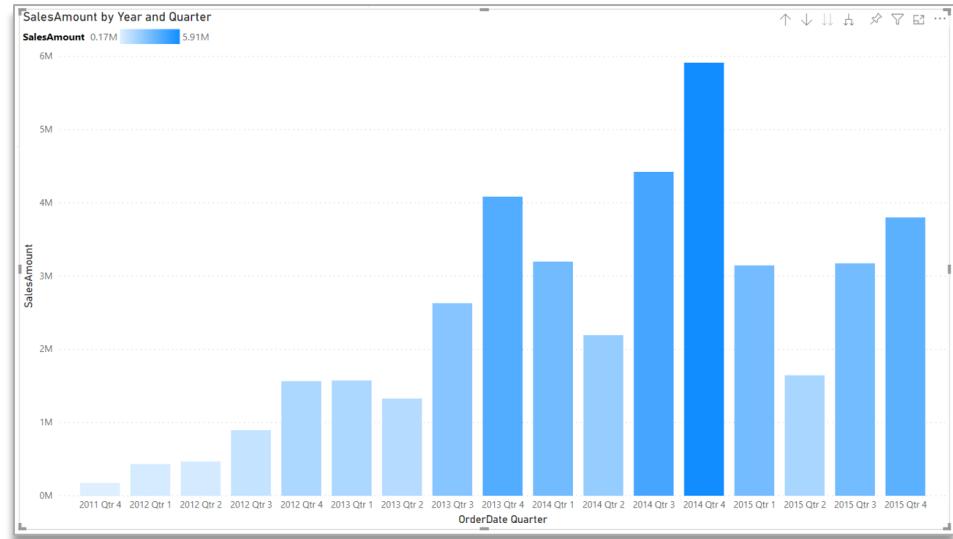
# Building a Report

## More reporting features - Gradient Legend

You can include a **legend** for data colors that have been conditionally formatted by color scale to help clarify the meaning of the colors.

You can use **conditional formatting** dialog in the **Data colors card** of the formatting pane to set a new rule to format your data along a color scale.

This will enable the **Legend** card in the formatting pane. You can **enable** the **gradient legend** by toggling it on, then format the legend as desired.

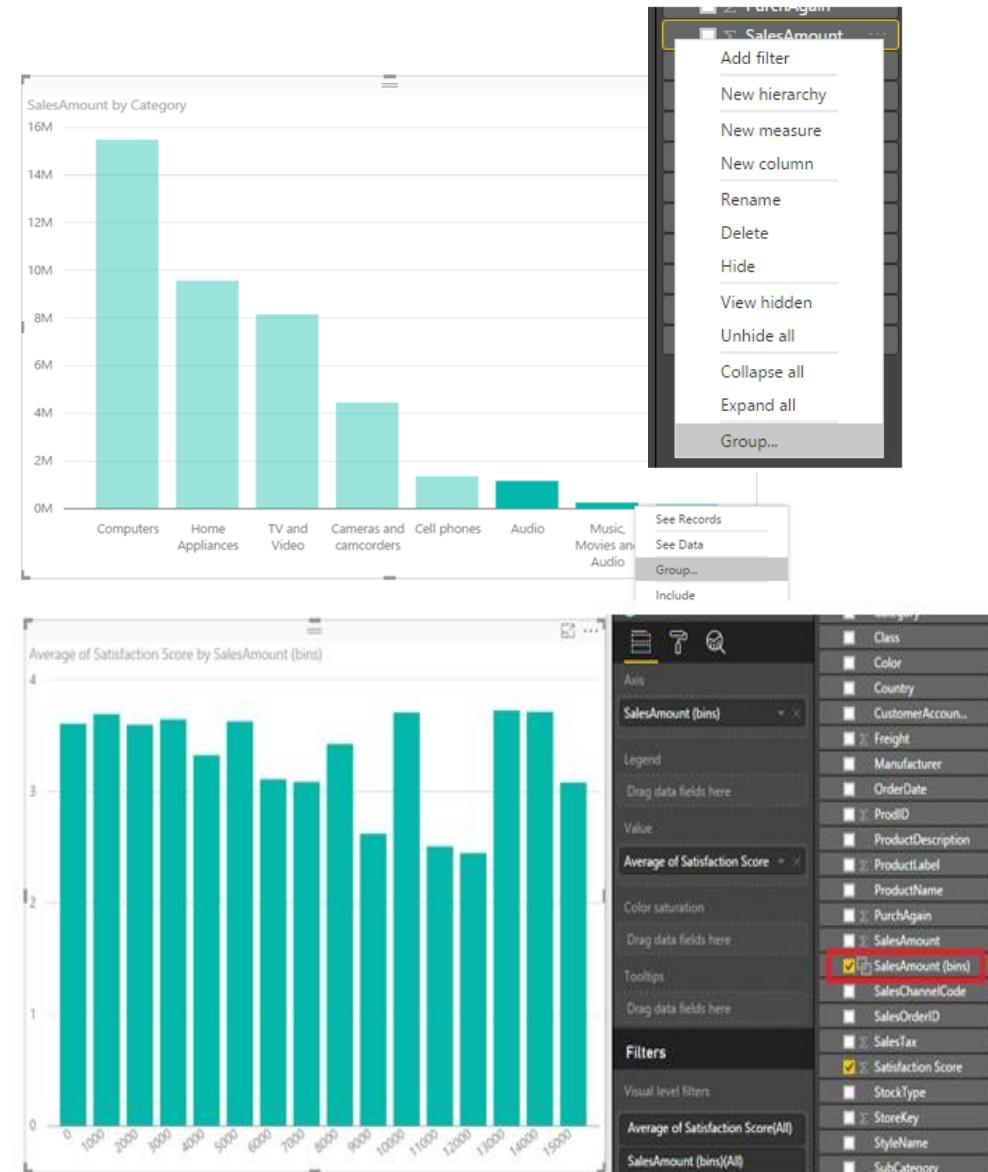
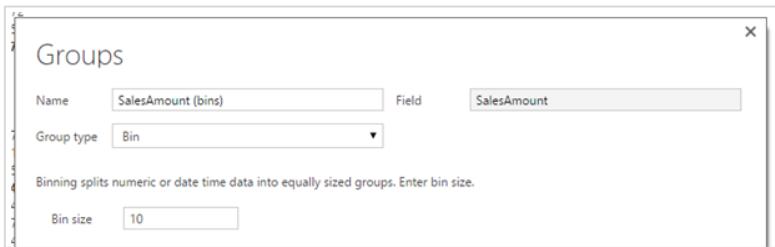


# Building a Report

More reporting features - Grouping and Binning

**Group** up points on your visual to easily explore and analyze related categories

**Bin** on numerical and time fields to create data categories



# Building a Report

## More reporting features - Clustering

**Clustering** lets you quickly find groups of similar data points in a subset of your data

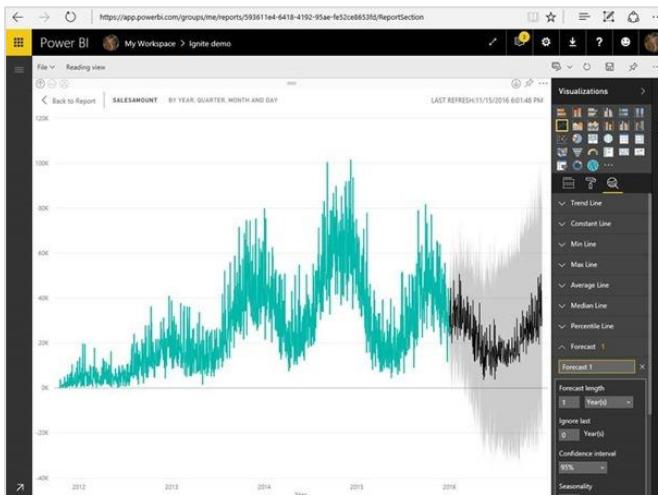


# Building a Report

More reporting features - Forecasting

You can use the **forecasting** feature on your line chart to do predictive analytics on your data.

**Built-in predictive forecasting** models to automatically detect the step (monthly/weekly/annually) and seasonality in your data to provide forecasting results.



# Building a Report

## More reporting features - Automatic Date

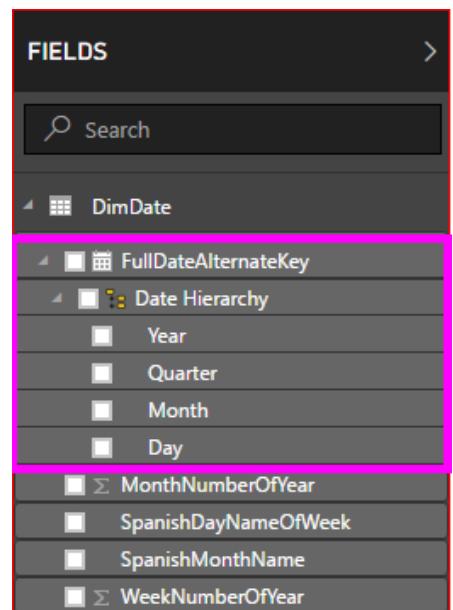
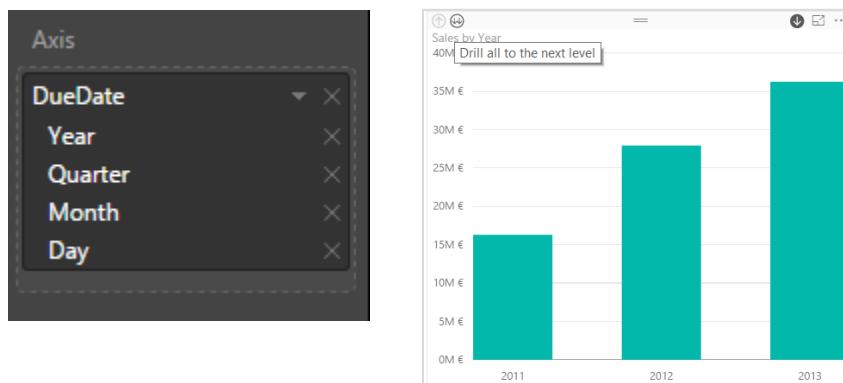
### Automatic detection of DateTime fields

DateTime Fields are **automatically expanded** when added to a visualization and **drilling** is enabled

The DateTime field **cannot** be connected to an **external table**

**Enable or disable** under Options and Data Load

Shows **date hierarchy** in the field list and allows **individual selection**

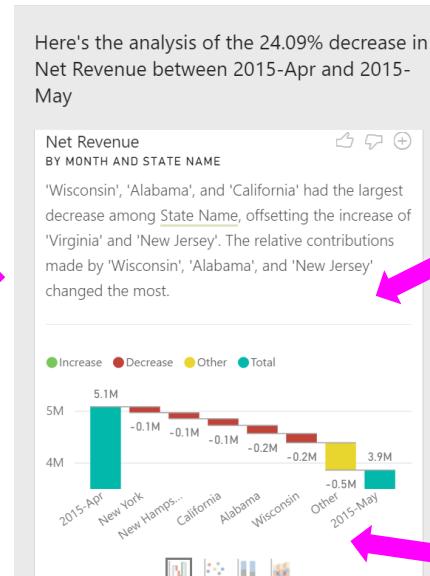
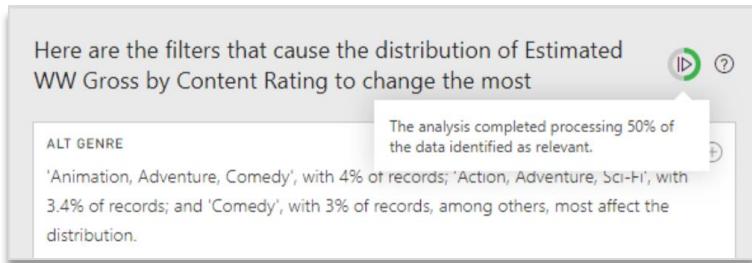
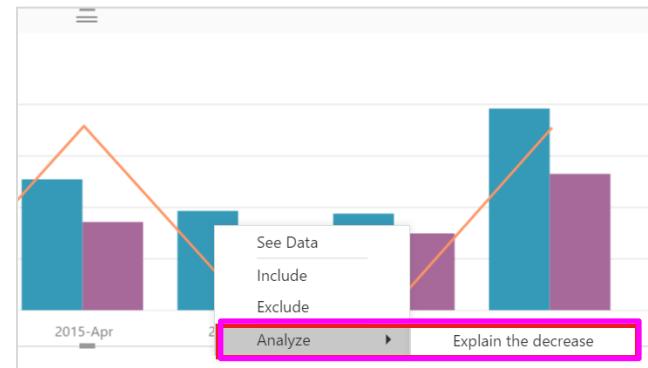


# Building a Report

More reporting features - Increase/Decrease Insights

Right click on a bar or a data point in a line chart

Ask us to explain why the data point increased or decreased compared to the data point before it.



machine learning algorithms

waterfall chart,  
scatter chart,  
stacked column  
chart or  
the ribbon chart

# Building a Report

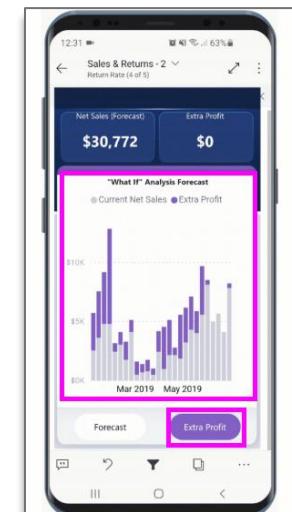
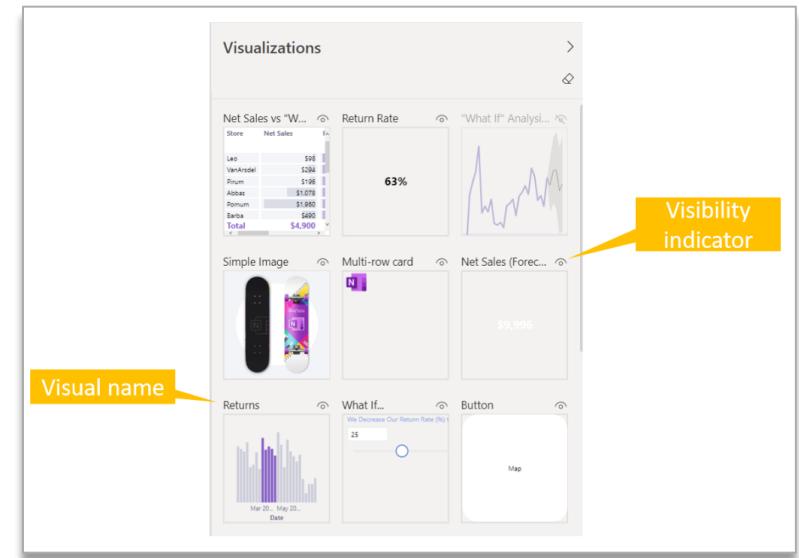
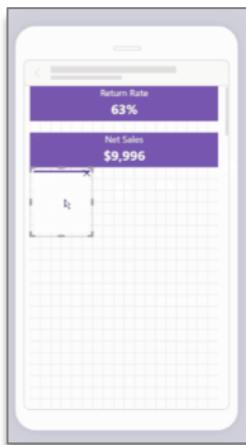
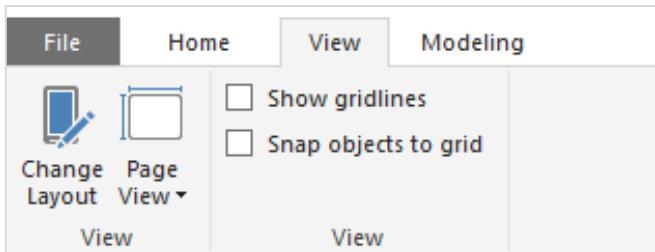
More reporting features - Mobile report layout

Create report **layouts specific to mobile** devices

**Visibility indicator** allows you to differentiate between hidden and shown visuals.

After publishing, anyone who views this report on their Power BI mobile app will see the **phone layout** of the report

The ability to **overlay visuals** makes it possible for you to use the same design techniques you use in **web layout** to create **beautiful, interactive, mobile-optimized reports**.



# Building a Report

## Row-Level-Security (Creation)

Define **roles and rules** within Power BI Desktop to restrict the data users see

**Supported for both Imported models and DirectQuery connections**

Rules (or filters) defined within roles **restrict data at the row level**

Rules are defined using **DAX**

The screenshot shows the Power BI Desktop interface with the 'Modeling' tab selected. In the top right corner, there is a 'Security' group containing icons for 'Manage Roles' and 'View As Roles'. A pink box highlights the 'Manage Roles' icon. Below the ribbon, the 'Manage roles' dialog is open. It has three tabs: 'Roles', 'Tables', and 'Tables Filter DAX Expression'. The 'Roles' tab shows a single role named 'United States'. The 'Tables' tab lists 'Customer Orders' and 'Date' tables. The 'Tables Filter DAX Expression' tab contains the DAX expression: `[Region] = "United States"`. A pink box highlights the 'Create' button in the 'Roles' section of the 'Manage roles' dialog, and a hand cursor is shown pointing at it.

# Building a Report

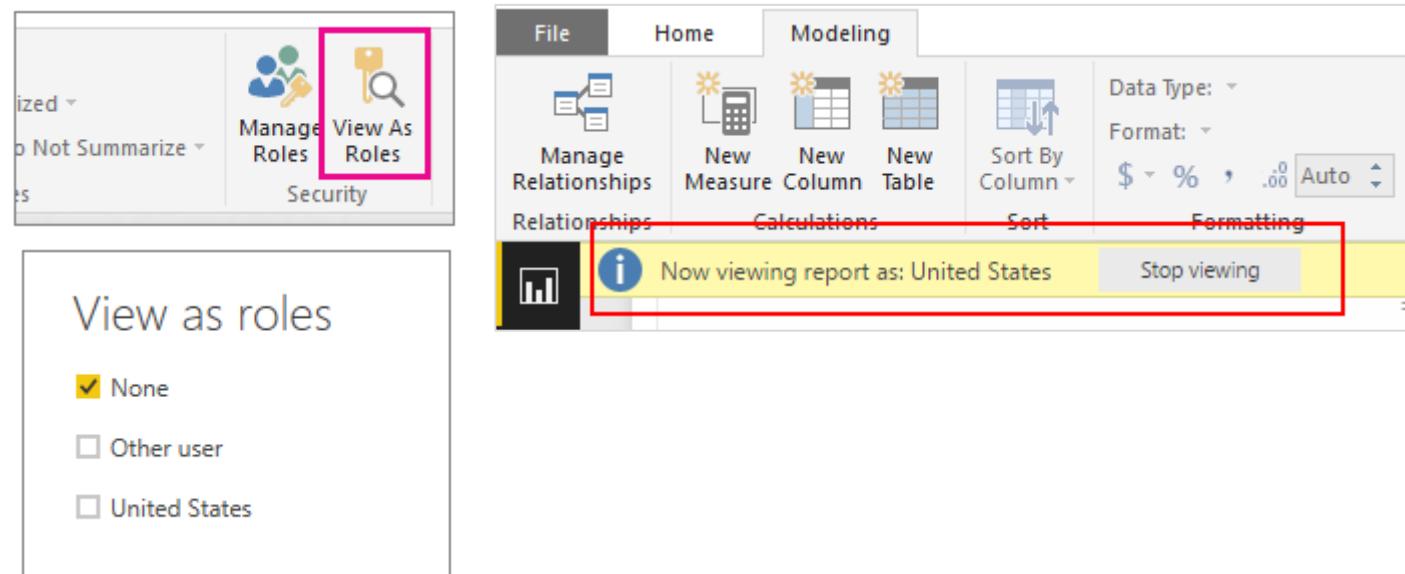
Row-Level-Security (Validation)

---

**Test the results** of the role within Power BI Desktop

---

Note that **users are assigned** to a role within the **Power BI service** after the solution is published

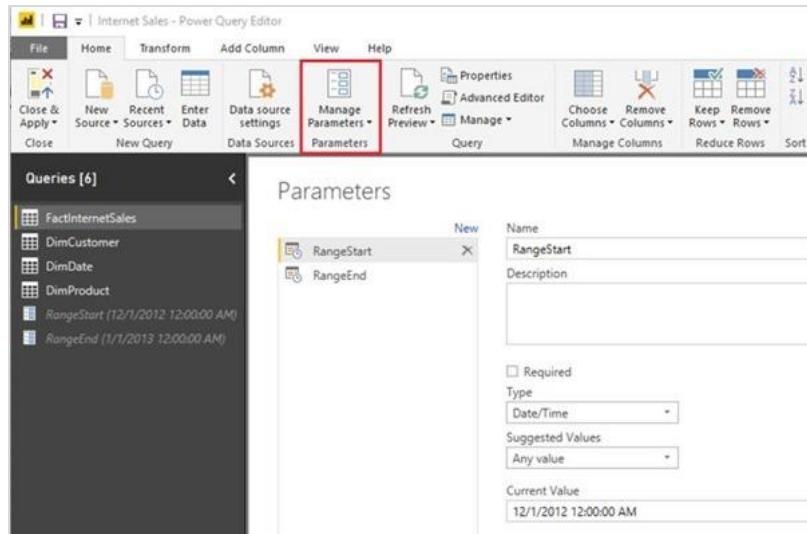


# Building a Report

Incremental Refresh – PRO + Premium

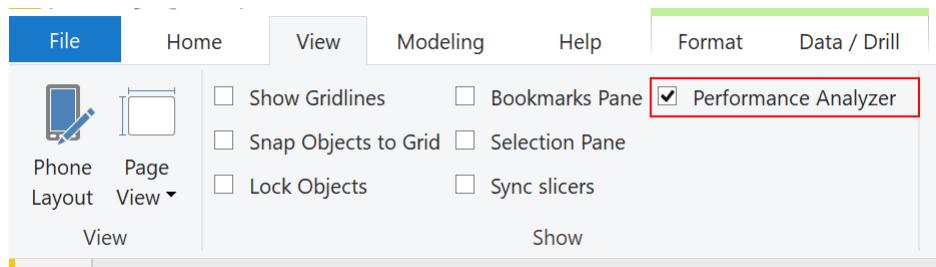
**Define refresh policy** in Power BI Desktop to determine how data is refreshed incrementally when published to Power BI Service

Filtering needs to be done **using Power Query date/time parameters** with the reserved names RangeStart and RangeEnd



# Building a Report

## Performance Analyzer



Measures the processing time (including the time to create or update a visual) required to update report elements.

To have Performance Analyzer begin recording, simply select **Start recording**.

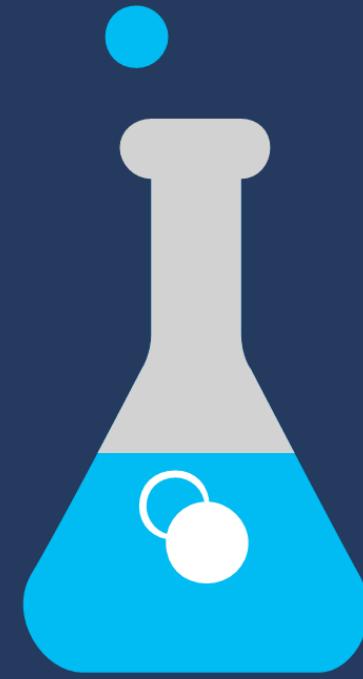
View and **Copy exact DAX query** for each visual.

The screenshot shows a Power BI report with two visualizations: a line chart comparing sales in UK and USA, and a bar chart showing the distribution of Sale Size by Class. An 'FILTERS' pane is visible below the charts. On the right, the 'PERFORMANCE ANALYZER' pane is open, displaying a table of processing times for various report elements. The table includes columns for Name and Duration (ms).

Name	Duration (ms)
Changed a slicer	-
Slicer	352
DAX Query	4
Visual display	151
Other	196
Copy query	
Key influencers	155
Header - Blue Square	329
Button P1	329
DAX Query	-
Visual display	175
Other	153
Copy query	
Button P2	328
Button P3	327
Button P4	326
Button P5	326
Button P6	325
Button P7	324
Button P8	323
Button P9	322
Button P10	321

# Power BI Desktop

- Exploring and Reporting



LAB

**Questions?**



# Knowledge Check

Name some of the features available in Power BI desktop to make your report interactive

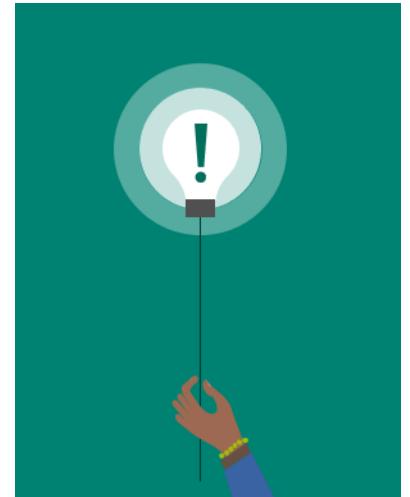
How can you import custom visuals to Power BI desktop?

# Lesson 7: Sharing

# Objectives

After completing this learning, you will be able to:

- Learn Power BI template files
- Learn how to publish data and reports to the cloud and connecting back to the cloud



# Sharing

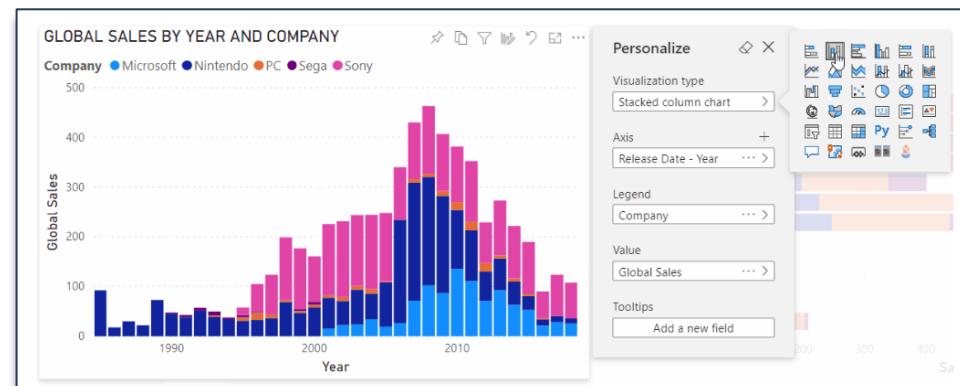
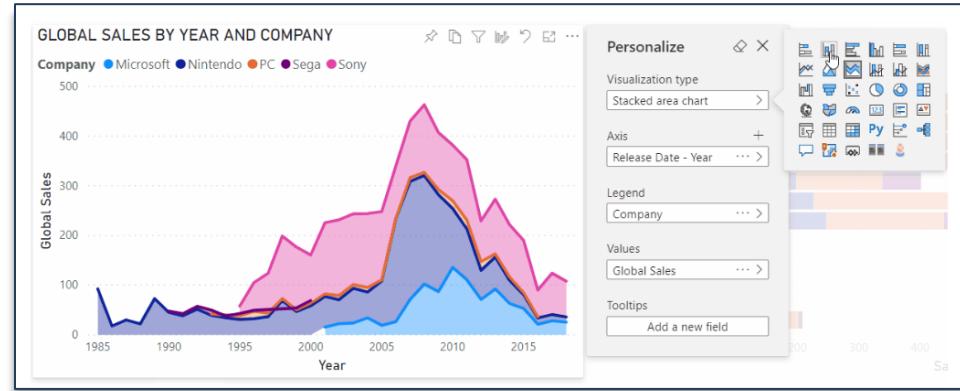
Personalize Visuals Experience for PBI Reports ( Preview )

Empower your **end-users** to explore and **personalize** visuals all within the consumption

## Consumer capabilities:

- Change the visualization type
- Swap out a measure or dimension
- Add or remove a legend
- Compare two or measures
- Change aggregations
- Capture their changes
- Share their changes
- Reset all changes to a report
- Reset all changes to a visual

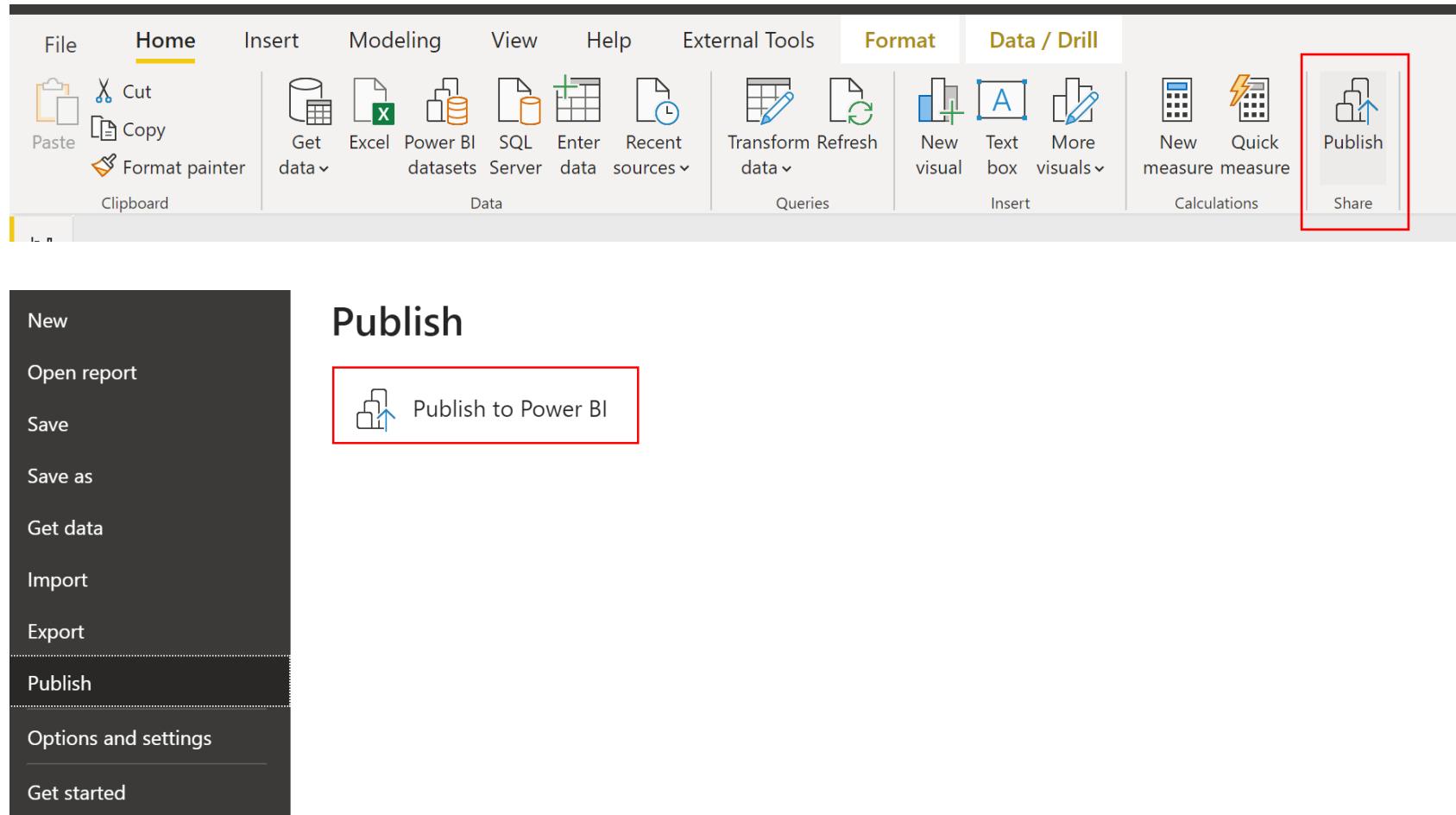
It can be **saved** as a **bookmark**.



# Sharing

Publish to the Cloud

The **data model, its data, and reports** are uploaded to the service



# Sharing

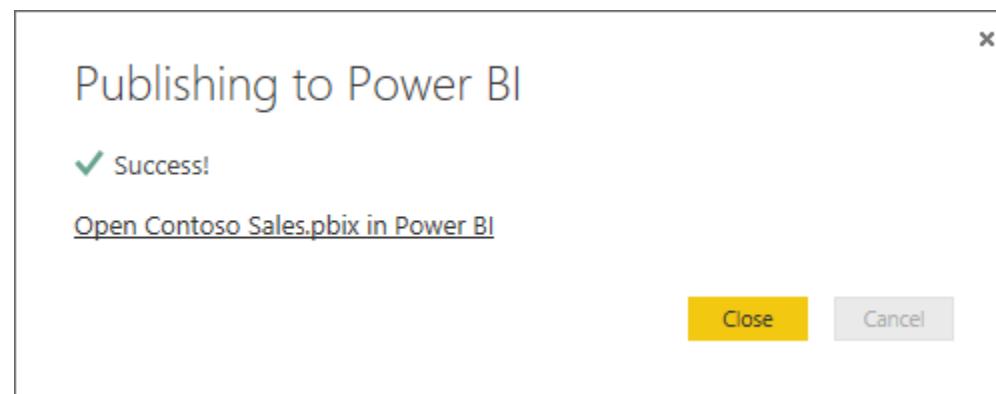
## Publish to the Cloud

Any **changes** made in the report after this or made in the service **will not be synced**

The **filename** is used for the **dataset name**

**Republishing replaces the existing dataset** (name must be the same)

In order to publish, you must **sign-in** to the service



# Sharing

Connecting back to the Cloud

Once a report is published, a dataset with the data model becomes available.

It is possible to **connect to it from Power BI Desktop** and use it as a support for reports

It is **not possible to change** the queries or the model, but you can **create DAX** measures (like LQ connections to SSAS Tabular model)

Can also **connect to dataflows** from Power BI Desktop

Get Data

Search

All

File

Database

**Power BI**

Azure

Online Services

Other

Power BI

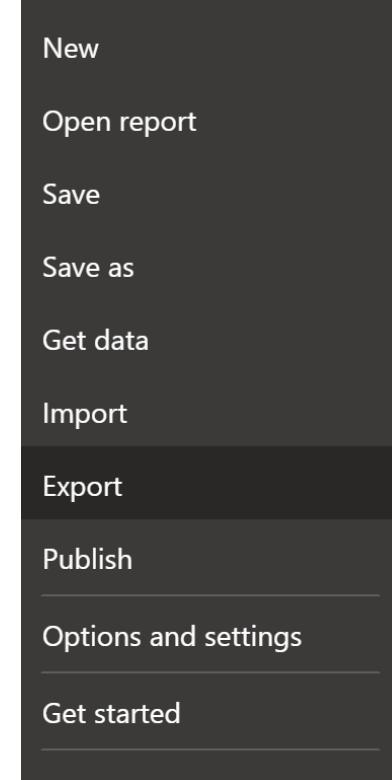
 Power BI datasets

 Power BI dataflows

# Sharing

Export report to PDF

Can **export** all visible report pages to **PDF** and then **print** or share via **email**



## Export



Power BI template



Export to PDF

# Sharing

## Power BI Template Files

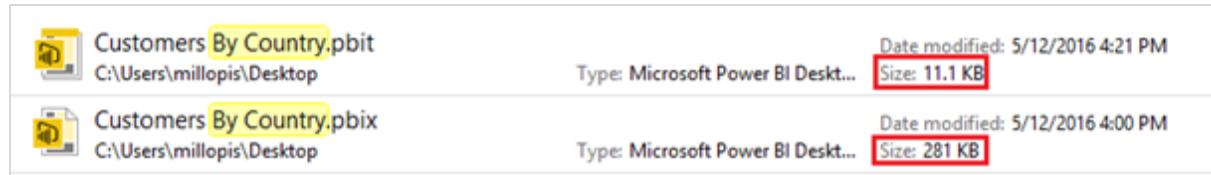
They are **generated from existing PBIX files** and contain **everything, except the data**

They are **Exported and Imported** (also with double-click on template file) **via the File Menu**

A template file has the **PBIT extension**

When a **template is instantiated**, the **data is re-read**, so the **credentials of the current user** can be used and **any parameters** will have to be **input**.

The template file will be **considerably smaller**



Several PBI templates on AppSource ( for ex Microsoft 365 Usage Analytics )

# Sharing

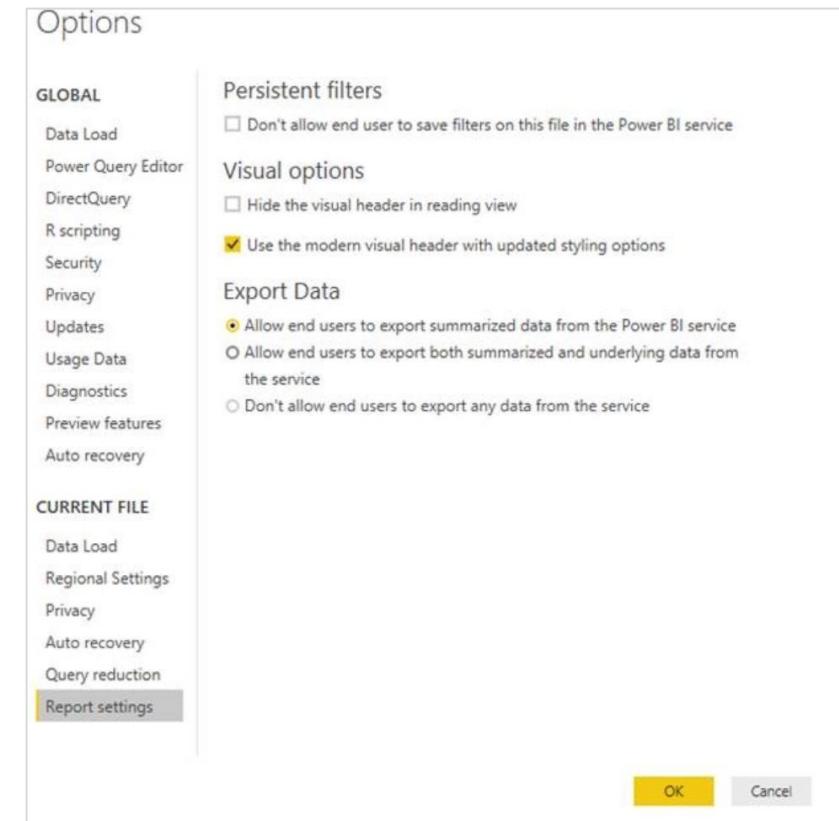
## Export data options

**Flexibility** to control what types of **data export options** are available for your consumers when the report is published to the Power BI service.

You can choose to:

- Allow **exporting** only of **summarized data** (this will be the new default for reports)
- Allow exporting of **summarized** and **underlying** data (this is the default for all old reports)
- Don't allow exporting of any data

You can set this in **Power BI Desktop** through the **options** dialog under **Report settings**.



# Demonstration

## Using Templates



**Questions?**



# Knowledge Check

What are some of the data connectivity modes in Power BI?

Name a few built-in transformations in Power Query editor.

What are calculated columns and calculated measures?

What is DAX used for?

Can you build your own visuals for Power BI reports?

What are some of the ways to share your Power BI reports?

# Module Summary

Describe Power BI Desktop Workflow

- Power BI report authoring process

Get Data

- Connect to data using different methods

Shape Data

- Transform data to meet business requirements

Model Data

- Create relationships and calculations

Visualize Data

- Visualize data

Sharing

- Share report with consumers

