



# Data Action Lab



**Enseignement univ.:** 50+ cours; **ateliers:** 40+; **projets:** 60+; **expérience combinée:** 35+ années.  
Co-entreprise pré-qualifiée à la liste des fournisseurs I.A. du GdC – EN578-180001/A (1<sup>ière</sup> bande).

Nouveau catalogue de formation approfondie disponible sur la toile au **data-action-lab.com**

**Combined experience:** 50+ university courses, 100+ corporate workshops, 60+ projects, 35+ years.  
Joint venture qualified for GoC A.I. Source List – EN578-180001/A (Band 1).

New advanced training catalogue available at **data-action-lab.com**



**Training and long courses**



**Workshops and short courses**



**Knowledgebase curation**



**Data labs**



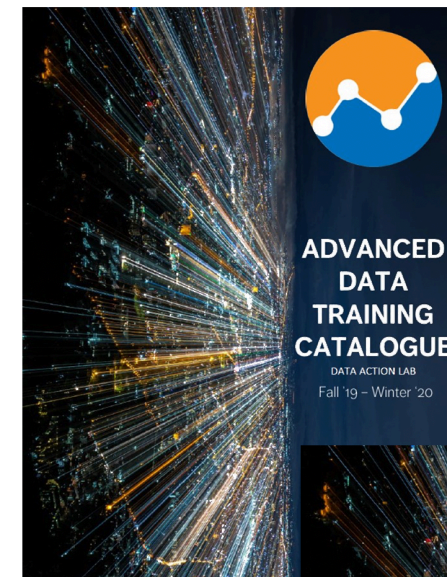


### Training Paths

- Data Novice
- Data Engineer
- Data Practitioner
- Data Scientist
- Data Manager
- Data Champion

### Training Learning Interests

- Visualization and Dashboards
- Introduction to Data Science
- Advanced Data Science
- Machine Learning Toolbox
- Spotlight on Classification
- Spotlight on Clustering
- Text Analysis
- Special Topics in AI and DS
- Hands on Data Analysis
- Data Strategy and Governance





- Business intelligence
- Data visualization design
- Data analytics and data science
- Data engineering
- Advanced statistics and machine learning
- Artificial and augmented intelligence
- Process and systems modeling
- Software implementation and integration







**Provide a space for data consumers, producers, practitioners, scientists and champions to make a place for themselves in the digital world.**



**Provide paths for education and enrichment for all these groups.**



**Keep pace with developments in the digital arena and keep Data Action Lab participants moving and aligned with these relevant developments.**



**Provide just-in-time learning opportunities for Data Action Lab members, focusing on their specific challenges and skillsets.**

# PBI-3: POWER BI – BEYOND THE BASICS

DATA ACTION LAB – POWER BI SERIES





# COURSE OVERVIEW

---

**Course #:** PBI-3      **Duration:** 0.5 day  
**Course Title:** Power BI Beyond the Basics

**Description:**

## **Head Start on Data Modeling**

- Where to use M and where to use DAX
- Best practice in setting up data relationships
- Columns vs Measures
- Overview of Quick Measures
- Distilling DAX
- Introduction to DAX
- Getting started, easy to use but useful DAX functions

## **Distilling DAX**

- Introduction to DAX
- Getting started, easy to use but useful DAX functions

# COURSE OVERVIEW

---

## Common Data Analysis Tricks

- Grouping
- Sorting
- Drill-through filtering
- Understanding time intervals
- Calculating with offset periods

## Simple Power BI “Hacks”

- Dynamically highlighting values
- Dynamic colour in tables
- Gradient colouring
- Dynamic cards

## Non-Standard Charts

- Introduction (Microsoft Store add-in charts)
- Overview of some of the better non-standard (Microsoft Store) charts



---

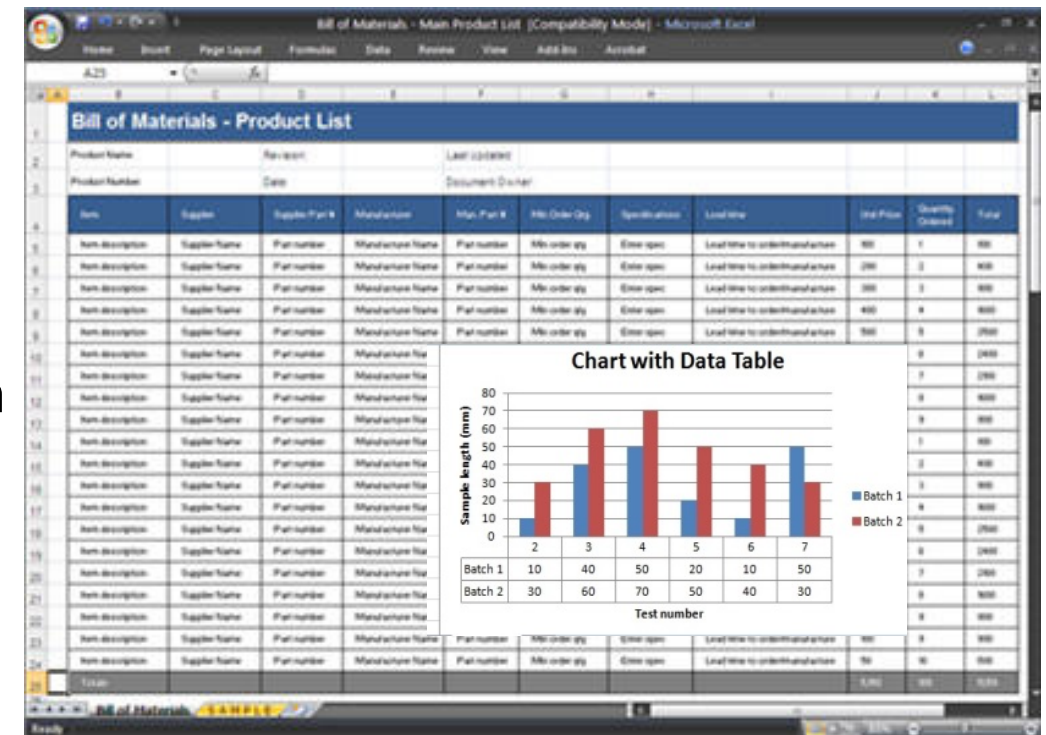
# INTRODUCTION

PBI-3: POWER BI – BEYOND THE BASICS

# OVERVIEW

## The past is **data-driven**:

- mostly Excel (or reporting tools like Cognos)
- mostly numbers, tables and non-interactive graphs
- distributed on desktop computers, by email, in PowerPoint presentation
- static, mostly backwards looking (lagging indicators)
- KPIs and dashboards were somewhat contrived

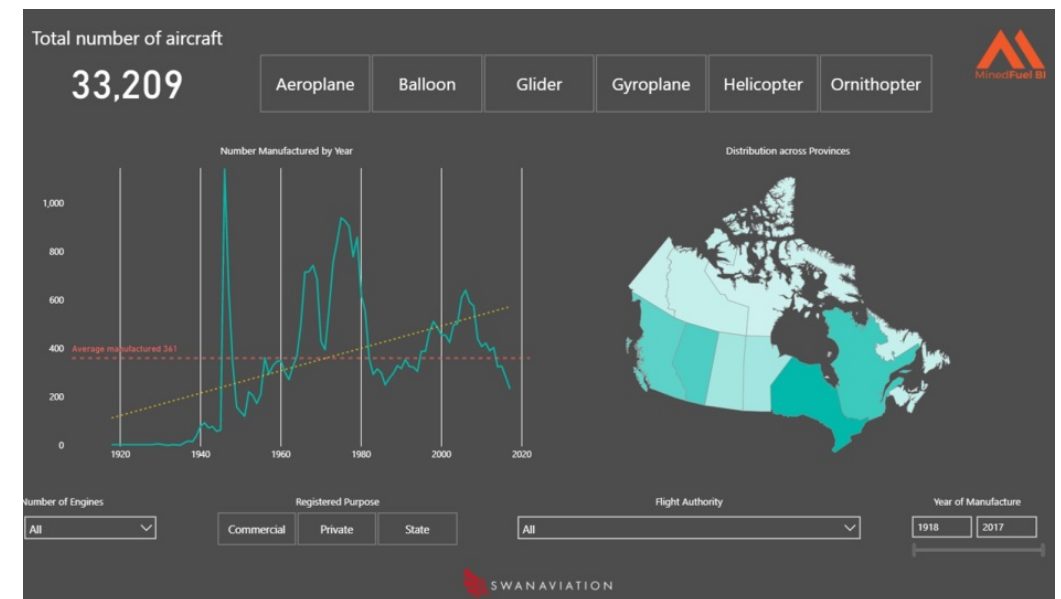




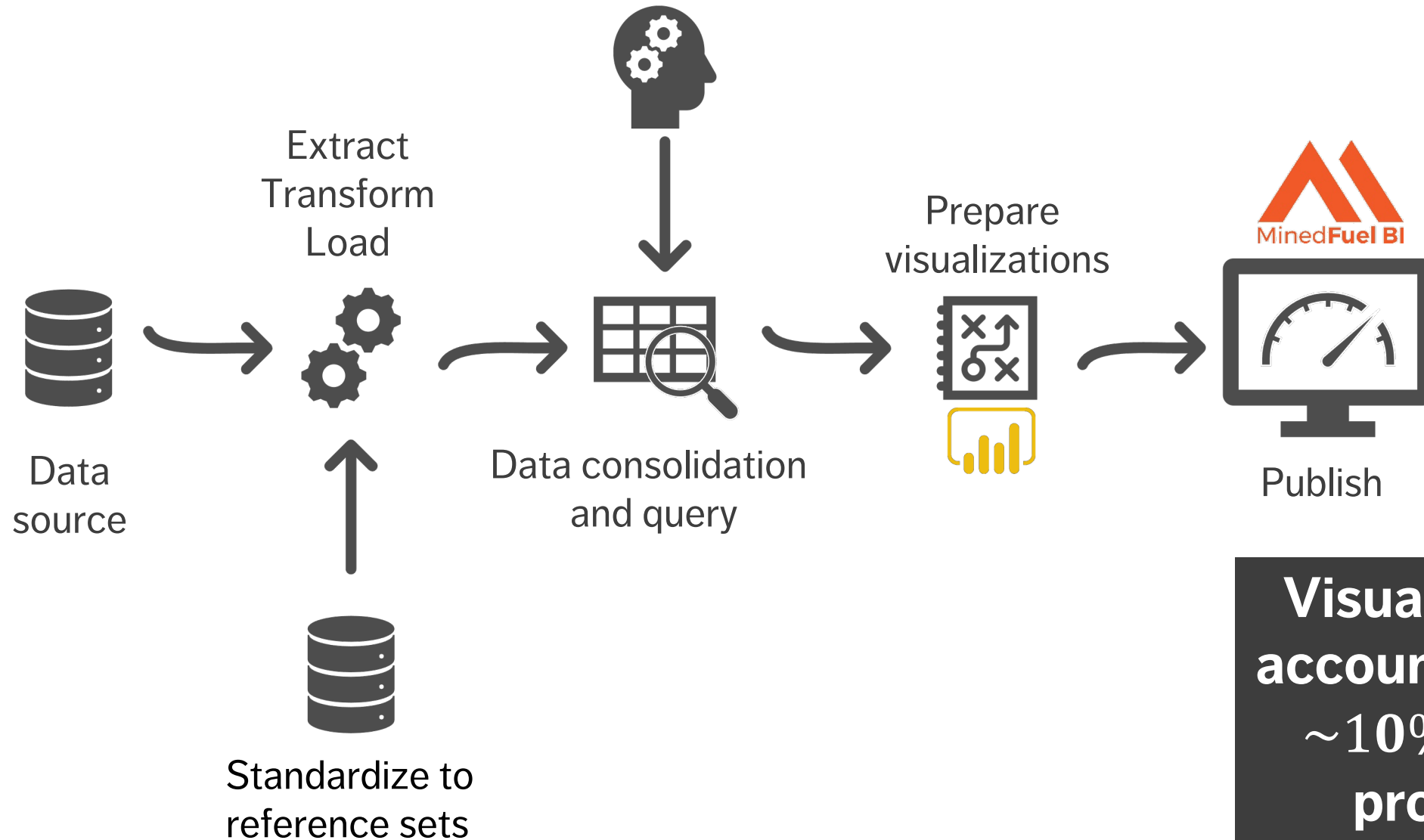
# OVERVIEW

## The future is **story-driven**:

- new tools: Power BI, R, Qlickview etc.
- mostly visualizations, occasional numbers and tables
- distributed on the web (internal and external)
- dynamic and both backwards and forwards looking (leading and lagging indicators)
- data is for everyone



# STRUCTURE OF DATA PROJECTS



**Visualizations  
account for only  
~10% of the  
process.**

# DEFINITIONS

In this course

- **Data Model:** the data that you pull in from your source (excel or database) and the schema that is associated with the data (i.e. the relationships and type of relationship)
- **Data Analysis:** useful and commonly used algorithms, tricks and tips to give you a head start in Power BI
- **Hacks:** useful but little used, unknown, unusual or sometimes blindingly obvious tricks that make you and your end users lives easier!!

# HEAD START ON DATA MODELING

PBI-3: POWER BI – BEYOND THE BASICS



# DATA MODELING – WHAT HAPPENS WHEN

We call the activities we do before visualization “**Data Engineering**”.

The three typical tools / approaches that we use for data engineering are:

1. SQL **before** data import (not included in this course)
2. M **during** data import (look at a high level in this course)
3. DAX **after** data import (go somewhat in depth in this course)

# M AND DAX

## What is M?

- M is the **data transformation engine** in Power BI
- M Query is a mashup query language used to query multiple sets of data sources
- M contains commands to transform data and can return the results of the query and transformations to the Power BI data model
- Normally, we use M Query to query data sources, clean and load data

# M AND DAX

## What is M (continued)?

- You can use M for any data preparation and data transformation tasks before loading the data into your model
- For example, instead of bringing 3 tables into Power BI, you can remove unneeded columns and merge them all together using M and create a single table to load into the model
- This can reduce the load and improve performance once the data model is loaded

# M AND DAX

## What is DAX?

- Data Analysis eXpression Language
- DAX is the common language used by SQL Server Analysis Services Tabular, Power BI, and Power Pivot in Excel
- DAX is the **analytical engine** in Power BI
- You use it once the data is loaded by creating custom columns, tables or measures, for instance
- Unlike M, it has some similarities to Excel functions
- However, DAX provides much more power than Excel formula (in many ways)



# DATA MODELING BEST PRACTICES

A lot of time in Power BI is spent on optimizing the Data Model.

Optimization is key in ensuring maximum performance and also to simplify the Power BI environment as much as possible for the designer

## **Best practices include:**

- use lookup tables (not usually done in Excel because we do not have relationships)
- avoid complex queries in Query Editor, if needed simplify data source (e.g. use views)
- when developing measures, start simply and then add complexity incrementally

# DATA MODELING BEST PRACTICES

## Best practices include (continued):

- use meaningful names for columns, tables and measures
- use a separate table for ALL measures
- remove unnecessary columns
- split date and time fields
- ... and thousands more!!!

We will look at some of these in detail at a later stage.

# COLUMNS VS MEASURES

In Excel, we can add calculated **columns**; this can also be done in Power BI, but we can also use **measures**.

**Calculated columns** (and tables) are:

- evaluated for each row in the table, immediately after 'Enter' is hit to complete the formula,
- then saved back into the model (so they take up space).

# COLUMNS VS MEASURES

**Calculated measures** are:

- evaluated when used in a visual or when the visual is rendered, and so
- not saved anywhere (aside from the cache).

Measures are generally used over calculated columns, but there are **trade-offs** with performance (report runtime vs. pre-processed), storage space, and the type of expressions that can be used.

Calculated columns are often used to **filter on the result** rather than just as a calculated result (for example in **slicers**).



# COLUMNS VS MEASURES – WHICH TO USE?

When do we use calculated columns over measures? Sometimes either is an option, but in most situations **the computation needs determine the choice**.

A calculated column needs to be defined to do the following:

- place the calculated results in a slicer, or in the axis of a chart, or use the result as a filter condition in a DAX query;
- define an expression that is strictly bound to the current row (for example, “Cost \* Volume” does not work on an average or on a sum of two or more columns);
- categorize text or numbers (for example, a range of values for a measure, a range of customer ages, such as 0–18, 18–25, etc.)

# COLUMNS VS MEASURES – WHICH TO USE?

Typically, measures are defined whenever a resulting calculation needs to be displayed in the values area in the plot area of a chart:

- for instance, in calculating the cost percentage on a certain selection of data.

# QUICK MEASURES

**Quick measures** can be used to quickly and easily perform common, powerful calculations.

A quick measure runs a set of DAX commands behind the scenes (no need to write the DAX code – it's done for you!) based on the input provided in the dialog box, then presents the results for use in the report/dashboard.

Best of all, the resulting DAX code that is executed by the quick measure can be viewed, which can help jump-start or expand one's DAX knowledge!

Let's open Power BI and take a look...

# DISTILLING DAX

PBI-3: POWER BI – BEYOND THE BASICS



# INTRODUCTION TO DAX

DAX is a collection of

- functions, operators, and constants
- that can be used in a formula, or expression, to calculate and return one or more values.

DAX helps create new information from data already found in the model.

**Great resource:**

- <https://docs.microsoft.com/en-us/power-bi/guided-learning/introductiontodax>

# INTRODUCTION TO DAX - RECAP

This formula's syntax includes the following elements:

`Total Sales = SUM(Sales[SalesAmount])`

- the measure name is `Total Sales`
- the operator “=” indicates the beginning of the formula; when calculated, it returns a result
- the DAX function `SUM` adds all of the numbers in the `Sales[SalesAmount]` column
- parenthesis ( ) surround an expression containing one or more arguments; all functions require at least one argument, which passes a value to a function
- the referenced table `Sales`
- the referenced column `[SalesAmount]` in the `Sales` table; with this argument, the `SUM` function knows on which column to aggregate

# EASY-TO-USE DAX FUNCTIONS – ARITHMETIC FUNCTIONS

The previous slide shows a standard use of SUM; DAX has the expected standard arithmetic expressions (SUM, DIVIDE, PRODUCT, MAX, MIN, AVERAGE, etc.)

But don't give in to the temptation to manually divide using a “/” operation, say (although everybody does and sometimes it is ok) – there is a failure option built into (for example) DIVIDE (or an alternate result, such as “0” for a “divide by 0” error).

```
DIVIDE(<numerator>, <denominator> [, <alternateresult>])
```

Spend a few minutes on a new tab playing around with these expressions; we will see them in action in a short while.

# EASY-TO-USE DAX FUNCTIONS – HANDLING ERRORS

If the dataset is well understood (or well-defined), there may not be a need to use error handling (with `IFERROR`).

In general, however, it is a very useful function to ensure that Power BI does not break down if (when) something goes astray:

```
IFERROR(<value>, <value_if_error>)
```

Try adding it to any and all of the DAX code you write as part of this course – upload a data set with known errors and see what happens.

# EASY-TO-USE DAX FUNCTIONS – CALCULATE & FILTER

CALCULATE is probably **THE MOST USEFUL** expression in DAX:

```
CALCULATE(<expression>, <filter1>, <filter2>...)
```

Even though we can filter within the CALCULATE expression, we often use the DAX FILTER expression as well:

```
FILTER(<table>, <filter>)
```

Tab “D1 – Calc & Filter” will walk us through an example.

## EASY-TO-USE DAX FUNCTIONS – MORE FILTERING

We are going to take a closer look at `SELECTEDVALUE` when we look at dynamic cards hacks later in the course (Tab “H3 – Dy Card”).

Briefly, it returns a value (e.g. a name) when a filter has selected ONE value only. This can be very useful especially when combined with variables:

```
SELECTEDVALUE(<columnName>[ , <alternateResult> ] )
```

## EASY-TO-USE DAX FUNCTIONS – MORE FILTERING

DISTINCT is very useful when creating lookup tables.

Use it to create a new column (or table) that only returns a set of distinct values:

`DISTINCT (<column> )`

**Try it:** create a new table that returns the list of distinct cities from the dataset.

# EASY-TO-USE DAX FUNCTIONS – COUNTING

**Counting** is one of the most critical functions in data analysis:

- COUNT returns all the items in the selected (filtered) column

`COUNT(<column>)`

- DISTINCTCOUNT returns all the **unique** items in the selected (filtered) column

`DISTINCTCOUNT(<column>)`

We will be using these in most of the examples throughout the course.



# DAX – DECLARING VARIABLES

Experienced programmers know how useful it is to **declare variables**.

**Variables** are simply labels given to commonly used functions.

We are going to declare one now to use in Tab “A4 – Date & VAR” (which is a beginner hack later on in the course)

The VAR **keyword** introduces the definition of a variable (as many as wanted).

The RETURN keyword defines the expression to return as the result. The variables are used inside the RETURN expression, which are replaced by **computed values**.

# DAX – DECLARING VARIABLES

For instance, the following code creates a new variable (DATEADD does what is expected; more details in the following section):

```
Snow Depth YoY% =  
VAR PREV_YEAR =  
    CALCULATE(  
        SUM( 'Data' [Snow Depth] ),  
        DATEADD( 'Data' [Relevant Date].[Date], -1, YEAR))  
RETURN  
    DIVIDE( SUM( 'Data' [Snow Depth] ) - PREV_YEAR, PREV_YEAR)
```

## PUTTING IT ALL TOGETHER

We now have all the pieces required (and more) to do a simple calculation.

**Example:** compute the Compounded Annual Growth Rate (CAGR) of the Snow Accumulation between 2 filtered sets of years, by following the instructions in Tab “D2 – CAGR”.

**Hint:** the formula needed for the calculation is

$$\text{CAGR} = \left( \frac{\text{Depth of snow in last selected year}}{\text{Depth of snow in first selected year}} \right)^{1/\text{Total number of years selected}} - 1$$

---

# COMMON DATA ANALYSIS TRICKS

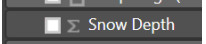
PBI-3: POWER BI – BEYOND THE BASICS

# GROUPING


Creating **Groups / Buckets** makes it easier to handle, among others:

- histograms
- manageable slicer controls

Fortunately, creating these in Power BI is *reasonably* straightforward:

1. Get your exercise .pbix and click on the ellipses (...) on the “Snow Depth” column in the report view (  ), then click on “New Group”
2. Select “Group Type” as “List”
3. In the ungrouped values hold down shift and select “0” to “9” and click “Group”

# GROUPING

4. The group title will be highlighted in blue, change this to “0 — 9” and hit enter
5. Repeat for the rest of the data set
6. You will now see a new group data field (  )
7. Create a new vertical chart with Axis as “Snow Depth (groups)” and Value as “Snow Depth” (an example is in “A1 – Sort & Groups” tab)

# GROUPING

The “Group By” list is the most controllable way of creating groups.

It is also possible to create bins in Power BI according to:

- a set bin size (e.g. 10cm), or
- the number of desired bins (e.g. 20)

Alternatively, it is possible to create a set of groups using one or many filters within a DAX expression if the need to be very specific arises (or if the query is too complex).

# SORTING

There are two ways to sort data in Power BI:

1. simply click on the ellipses on any chart and this will bring up the sorting options, or
2. to force a custom sorting order, use “Sort by Column” in the Modeling tab

For details and exercise, go to the “A1 – Sort and Groups” tab in the course .pbix file,



# DRILL-THROUGH FILTERING

**Drill-through filtering** can be very useful, but it does require educating the end user.

It requires at minimum 2 tabs: the first has a graph that is “drilled” and the second is a tab that contains the filtered “drilled” data.

Go to Tab “A2 – Drill example” (this is a pre-populated tab).

In the Filter tab on the right, drag 2 columns – “City” and “Snow Depth Groups” to the “Drillthrough filters” section. You will see an arrow icon now appear on the top right of the page

# DRILL THROUGH FILTERING

Go to Tab “A1 – Sort & Groups”. On the “Average of Snow Depth by City” graph, right click one of the bars, a menu will popup with “Drillthrough” on the bottom. Click on that and the tab “A2 – Drill example” will appear – click on it.

See that it takes you to that tab but applies the filter. Hold down control and click the arrow to go back.

Do the same for the histogram graph.

## UNDERSTANDING DATE AND TIME – “INTERVAL” TO DATE

Using “to date” (year or month) calculations yields different results than just using “sum” of a column in a chart value field.

We can see this behavior in Tab “A3 – YTD”.

The chart on the left “Snow Depth by Year (2003) and Month” simply shows the sum of snow depth for each month in 2003.

The chart at the top right does a simple annual sum of snow depth.

The chart at the bottom right does a Year to Date total (TOTALYTD) calculation.

## UNDERSTANDING DATE AND TIME – “INTERVAL” TO DATE

- Click on “2013 January”, you will see that both of the charts on the right show 273.
- Now click on “2013 February”, you will see that the YTD graph is now cumulative while the top graph (sum) just shows the month total
- Same applies for Month (TOTALMTD) and Quarter (TOTALQTD)
- Exercise – try a TOTALMTD or TOTALQTD calculation in a blank tab

# UNDERSTANDING DATE AND TIME – PREVIOUS PERIODS

Recall the code that was used for declaring Variables in Tab “A4 – Date & VAR”:

```
Snow Depth YoY% =  
VAR PREV_YEAR =  
    CALCULATE(  
        SUM( 'Data' [Snow Depth] ),  
        DATEADD( 'Data' [Relevant Date].[Date], -1, YEAR))  
RETURN  
    DIVIDE(SUM( 'Data' [Snow Depth] ) – PREV_YEAR, PREV_YEAR)
```

# UNDERSTANDING DATE AND TIME – PREVIOUS PERIODS

In the calculation in the previous slide, we select a previous period (previous year).

By far the most reliable way of doing this (for any period plus or minus, day, week, month, quarter, year, etc,) is to use DATEADD.

Look at Tab “A4 – Date & VAR” and play around with the previous period (as per instructions).

# SIMPLE POWER BI “HACKS”

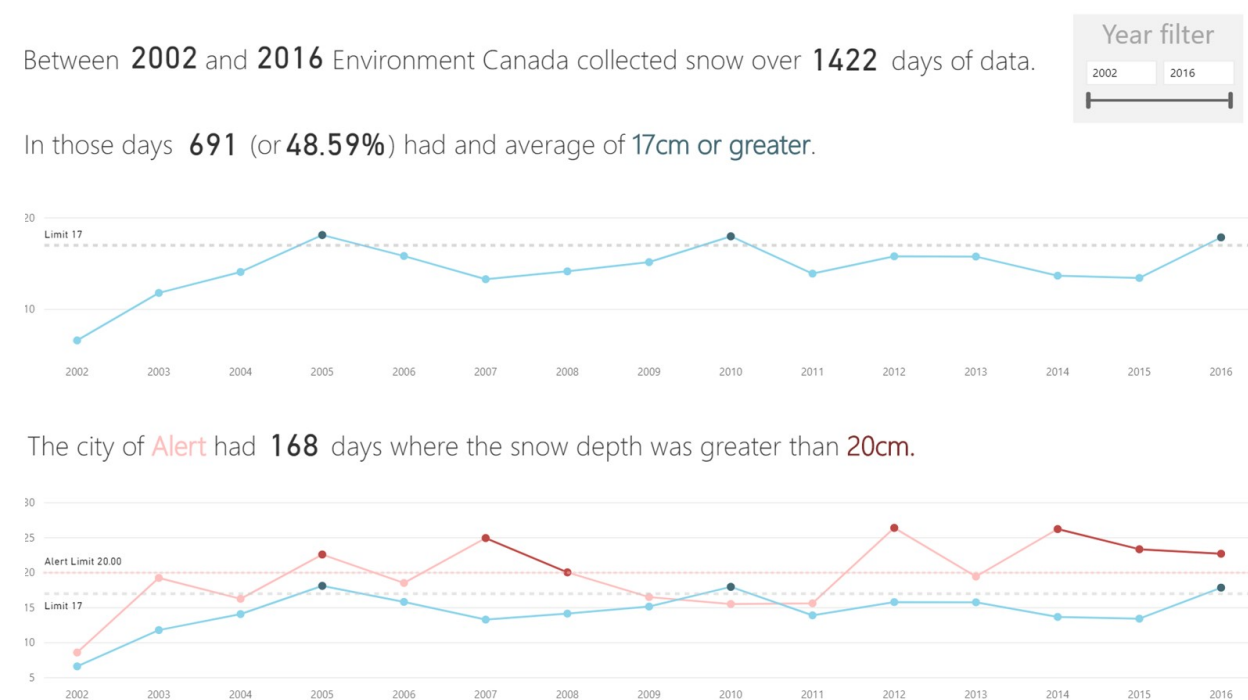
PBI-3: POWER BI – BEYOND THE BASICS

# DYNAMICALLY HIGHLIGHTING VALUES

Highlighting specific values can be useful, especially in line charts.

This can also be done dynamically (as the data changes, so does the highlighting), by creating a measure that contains some **comparison logic**.

Follow the instructions in the tab “H1 – HD” in the course .pbix file. Don’t forget to drill down the date hierarchy in the charts.





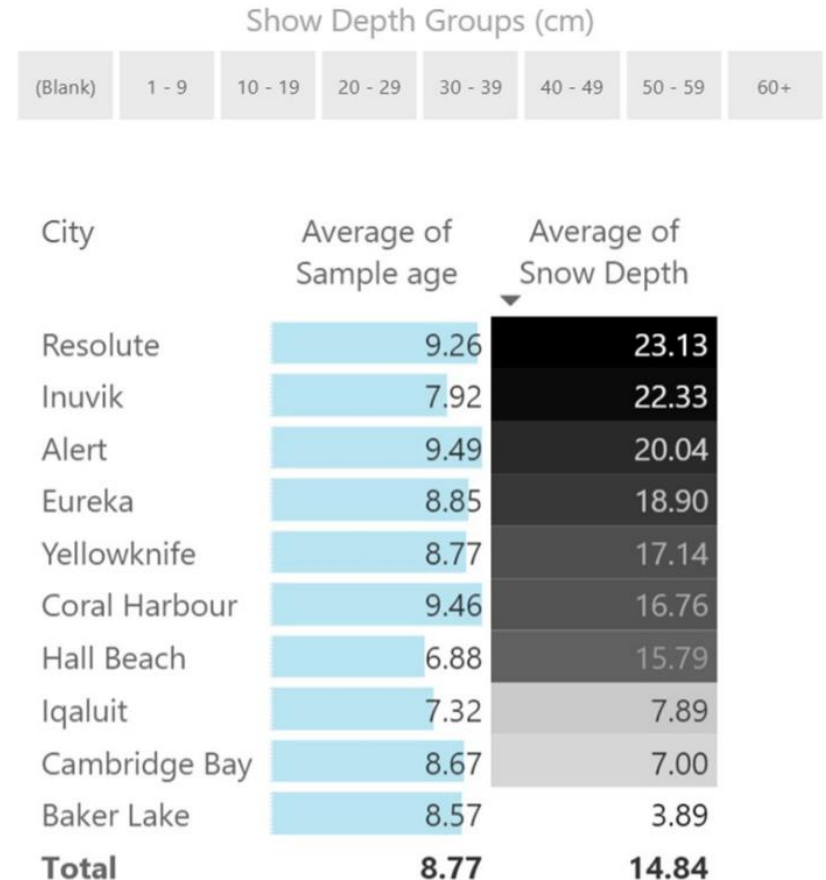
# DYNAMIC COLOUR IN TABLES

Tables can get very dense.

It is nice to be able to dynamically highlight values as filters / slicers are chosen.

This can be done without the need to write any DAX code.

See “H2 – Colour Table” in the course .pbix file.



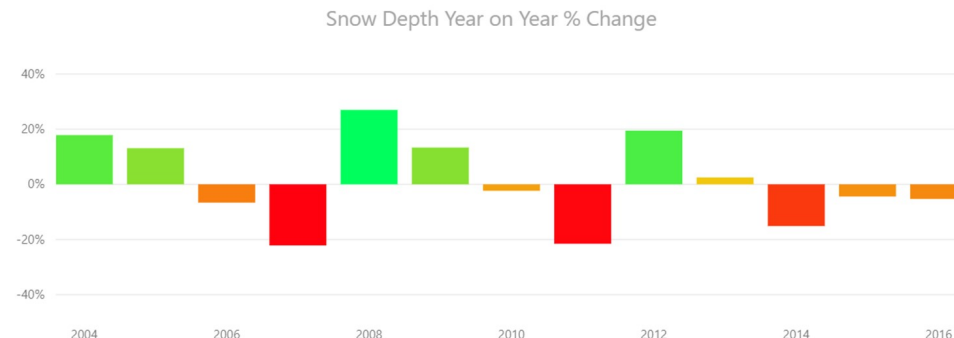
# GRADIENT COLOURING

Gradients can be added to a chart based on max / min value (e.g. green to red).

Work through an example in Tab “A3 – Date & Var”).

Once you have built the chart based on the guide do the following to colour it:

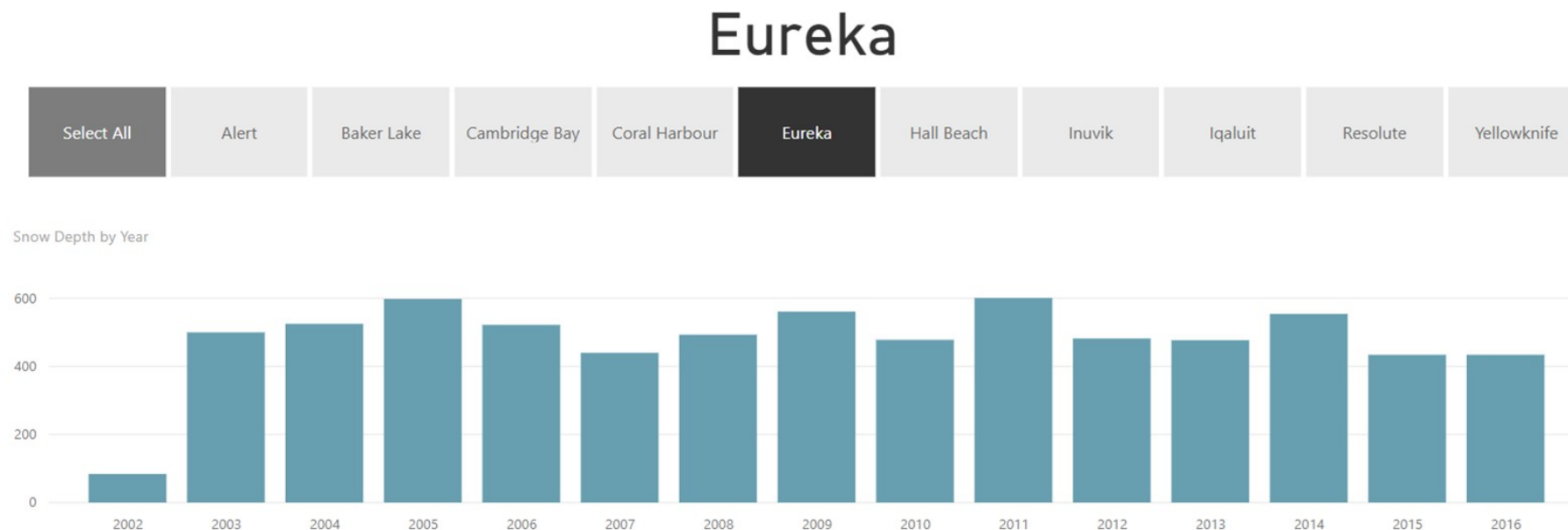
- drag the measure you created, “Snow Depth YOY%” to the Colour saturation section
- in the formatting tab, select “Data colors”, then select “Diverging ON”
- in the custom colours select Min, Center and Max colours (such as red – orange – green)



# DYNAMIC CARDS

Sometimes we want to be REALLY obvious about what values are selected. We can place extra emphasis by creating a dynamic card that shows the current selection(s).

Go to Tab “H3 – Dy Card” to work through an example.



---

# NON-STANDARD CHARTS

PBI-3: POWER BI – BEYOND THE BASICS

# DOTPLOT BY MAQ SOFTWARE

**Dotplots** displays data bubbles on an X/Y axis, distributed over a set of values.

They are very flexible and allow for compact display of data (MAQ charts: all good).

Used in Tab “A2 – Drill example”.



**Synoptic Panel** displays **maps** (images, not necessarily geographical maps), whose parts (**areas**) can be assigned “meaning”.

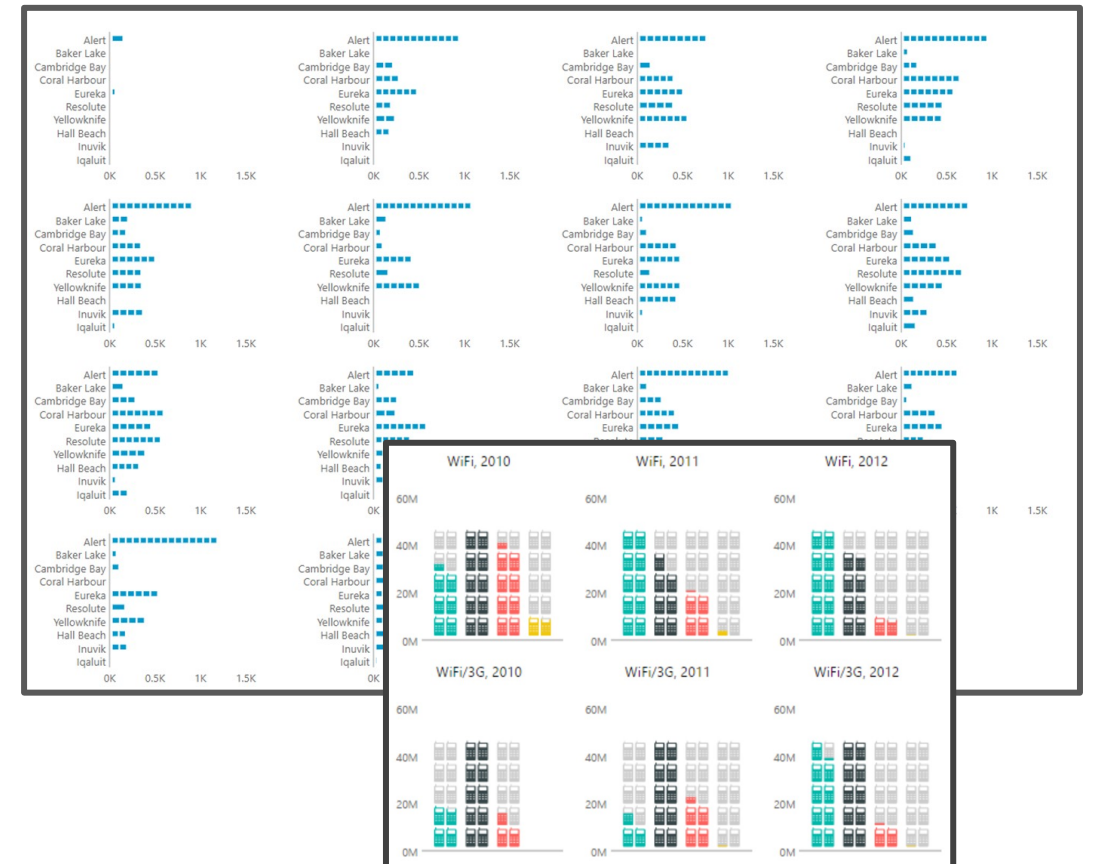
It's a good tool for processes, floor maps, custom geographic maps, etc.

Maps can be designed with a vector graphic editor or the companion tool **Synoptic Designer**.



# INFOGRAPHIC DESIGNER BY MICROSOFT

**Infographic Designer** “looks” like it is just used for creating charts with custom graphics, but it is **ESPECIALLY** useful for multi-slicing data (by column, by row, etc.).



# TEXT FILTER BY MICROSOFT

**Text Filter** is another Microsoft tool that is REALLY simple; both its functionality and performance are fantastic!

Use it in exploratory reports to allow users to rapidly search text.

## Access to Information and Privacy (ATIP) search

You have currently selected **28,711** requests totalling **6,597,612** pages of information

Range of dates available

11/1/2015 12/1/2017

### Keyword Searches

Search organization

Search Request #

Search summary description

Search Disposition

Organization
Administrative Tribunals Support Service of Canada
Agriculture and Agri-Food Canada
Asia-Pacific Foundation of Canada
Atlantic Canada Opportunities Agency
Atomic Energy of Canada Limited
Business Development Bank of Canada
Canada Border Services Agency
Canada Council for the Arts
Canada Deposit Insurance Corporation
Canada Economic Development for Quebec Regions
Canada Foundation for Innovation
Canada Mortgage and Housing Corporation
Canada Revenue Agency
Canada Science and Technology Museum
Canada-Newfoundland and Labrador Offshore Petroleum Board
Canada-Nova Scotia Offshore-Petroleum Board
Canadian Air Transport Security Authority

Date	Request #	Summary EN	Disposition EN
12/1/2015	A-2015-00324	Final report from investigation # 1410-2-2015-xx	
1/1/2016			
5/1/2016			
7/1/2016			
8/1/2016			
11/1/2016			
1/1/2017			
1/1/2017		Nothing to report	
1/1/2017	A-2016-01598	Please provide a list of all offices used by your organization in: Yukon territories, the North-West Territories and Nunavut.	
3/1/2017	A3-2016-00001		
3/1/2017	A13-2016-00001	Policies for travel and hospitality expenses incurred by the federally appointed commissioners, and for the reimbursement thereof. If the Commission has no such policies, please indicate whether it follows the Treasury Board's Travel Directive and Special Travel Authorities.	
6/1/2017	A-2017-00040	I would like the following two documents: 2017-01-17 POTENTIAL IMPACTS OF CARBON PRICING ON THE PRIMARY AGRICULTURAL SECTOR 226975 MINISTER MACAULAY 2017-03-08 AGRICULTURE AND AG RI-FOOD CANADA (AAFC)'S ANALYSIS ON THE POTENTIAL IMPACTS OF CARBON PRICING	
8/1/2017			

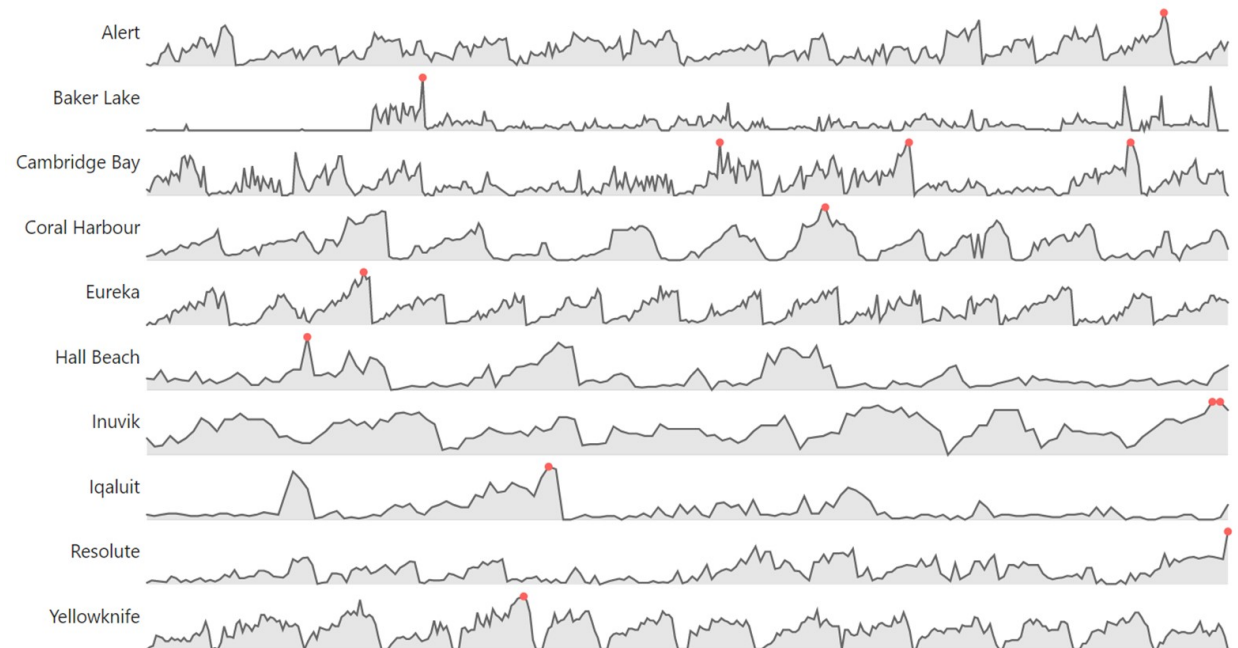


# SPARKLINE BY OKVIZ

**Sparklines** were introduced by Tufte; they are quite useful for summarizing a lot of data (usually time series data) in a small space.

There is a reasonable option menu (colours, highlighting, shading, etc).

It is easy to use, but does lack some labeling options.





# Questions?