DAX vs Power Query M

Brendan Clarke, NHS Education for Scotland, brendan.clarke2@nhs.scot

30/07/2024

KIND Learning Network

Welcome

- this session is for 🌶 Power BI beginners
- we’ll get going properly at 15.05
- you’ll need Power BI Desktop and this sample dashboard to follow along
- if you can’t access the chat, you might need to join our Teams channel: tinyurl.com/kindnetwork
- you can find all the session materials at tinyurl.com/kindtrp

The KIND network

- a social learning space for staff working with **k**nowledge, **i**nformation, and **d**ata across health, social care, and housing in Scotland
- we offer social support, free training, mentoring, community events, …
- Teams channel / mailing list

KIND
Learning Network

Session outline

- about DAX and PQM
  - DAX and PQM vs Excel formulas
- distinctive features
  - query steps (PQM)
  - filter context (DAX)
- applications and best practice
- feedback and resources

Setup

- Power BI desktop
- download and open this sample dashboard
    - three datasets, brought in from the web with PowerQuery
    - several calculated columns

About DAX and PQM

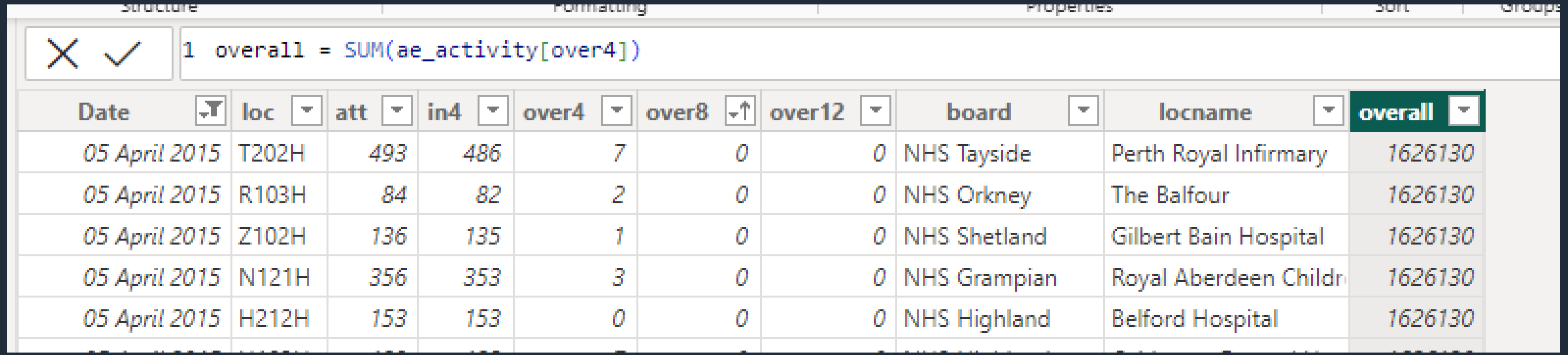
- found in Excel and Power BI (and in Microsoft’s SQL products)
- DAX (Data Analysis Expressions)
  - Excel: PowerPivot
  - Power BI: Measures and calculated columns
- PQM (Power Query M)
  - Excel: PowerQuery and various `Get Data` tools
  - PowerBI: various data loading tools and `Tranform data`

Different applications

- DAX = summarising/analysing data
- PQM = loading/transforming data

DAX vs Excel

- there are plenty of apparent similarities with Excel
  - broadly, functional approach
  - similar/identical function names
  - similar syntax in some places
- calculate a column `overall = SUM(ae_activity[over4])` in DAX
  - like Excel, this sums the entire `over4` column, rather than each row



```
1 overall = SUM(ae_activity[over4])
```

Date	loc	att	in4	over4	over8	over12	board	locname	overall
05 April 2015	T202H	493	486	7	0	0	NHS Tayside	Perth Royal Infirmary	1626130
05 April 2015	R103H	84	82	2	0	0	NHS Orkney	The Balfour	1626130
05 April 2015	Z102H	136	135	1	0	0	NHS Shetland	Gilbert Bain Hospital	1626130
05 April 2015	N121H	356	353	3	0	0	NHS Grampian	Royal Aberdeen Childr	1626130
05 April 2015	H212H	153	153	0	0	0	NHS Highland	Belford Hospital	1626130

KIND Learning Network

NHS Education for Scotland

PQM vs Excel

- PQ really looks like Excel
    - familiar tools - renaming/removing columns, filtering
    - evolved tools - like `Split Column`
- PQM is much less like Excel formula language than DAX

Appearances mislead

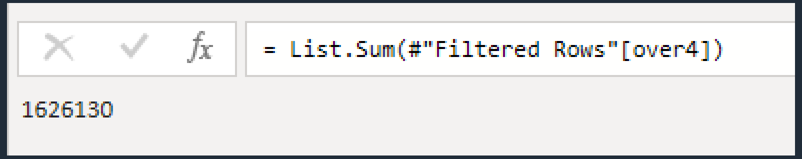
- try adding another column to the Excel formula, and to the DAX
  - e.g. `overall = SUM(ae_activity[over4], ae_activity[over12]))`
  - ✓ Excel is perfectly fine with this
  - ✗ but DAX's `SUM` function falls over
- for PQM, totally different approach required to Excel

Input in DAX

- DAX takes structured references to columns and tables (no `A3`)
  - `overall = SUM(ae_activity[over4])` sums all the values in the over4 column
  - `table[column]` - so this is the `over4` column in the `ae_activity` table

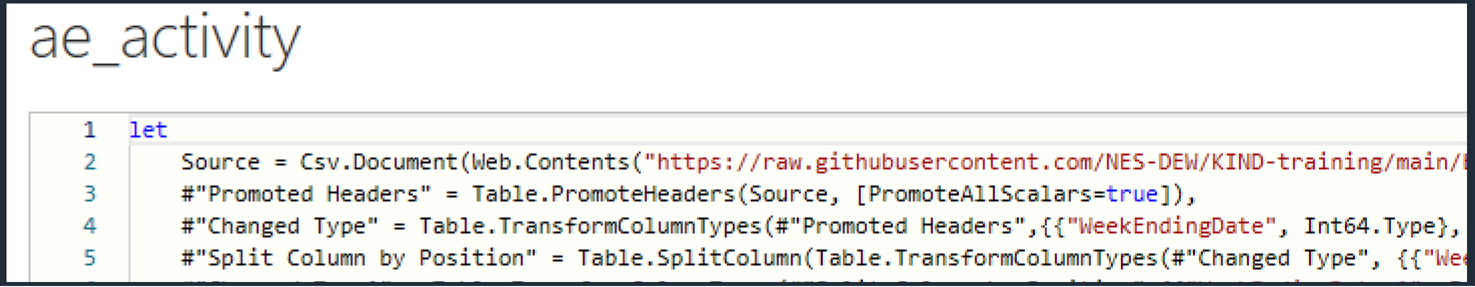
Input in PQM

- PQM works on **query steps**, with the **#step name** (and columns/tables) as input
  - `= List.Sum(#"Filtered Rows"[over4])` would sum all the values in the over4 column



```
= List.Sum(#"Filtered Rows"[over4])
1626130
```

  - takes the `#Filtered Rows`query step, and sums its `over4` column
  - that new query step will be called `#Calculated Sum` (but we could edit that)
- this is unusual, but gives PQM users a tweak-able history of their data transformation with undo/redo
- try looking at the advanced editor in PQM to see what PQM really looks like



ae_activity

```
let
    Source = Csv.Document(Web.Contents("https://raw.githubusercontent.com/NES-DEW/KIND-training/main/
    #"Promoted Headers" = Table.PromoteHeaders(Source, [PromoteAllScalars=true]),
    #"Changed Type" = Table.TransformColumnTypes(#"Promoted Headers",{{"WeekEndingDate", Int64.Type},
    #"Split Column by Position" = Table.SplitColumn(Table.TransformColumnTypes(#"Changed Type", {{"We
```

KIND Learning Network

NHS Education for Scotland

Filter context

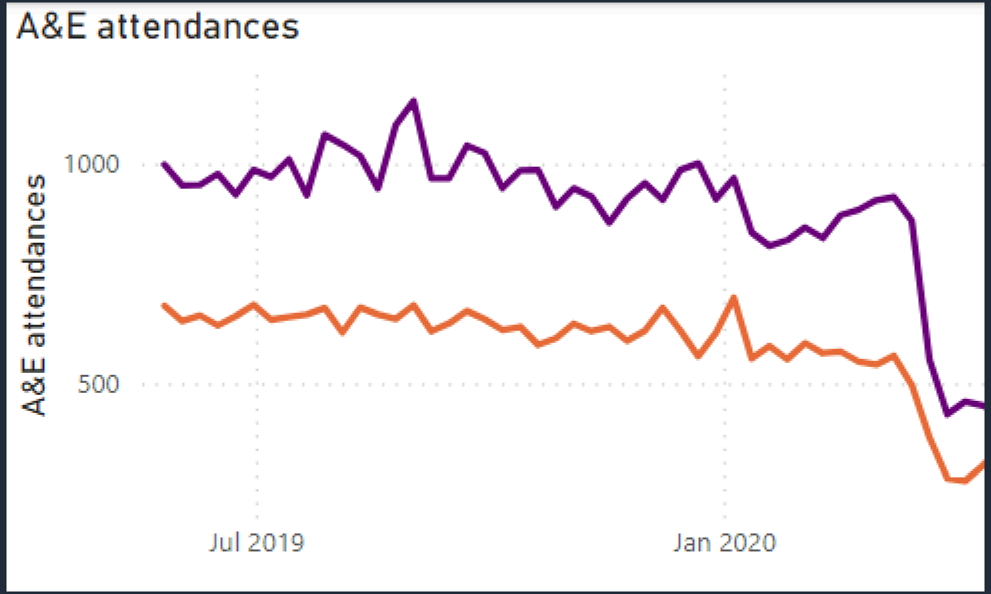
- there is considerable overlap between DAX and PQM
  - example: DAX’s calculated columns replicates functionality in PQM (and Excel)
- to show the DAX-specific part of the story, we’ll need to make a measure
- measures are responsive summaries of our data - when a user twiddles the dashboard, they’ll change
  - or, *measures respond to the filter context*

Make a measure

KIND Learning Network

NHS Education for Scotland

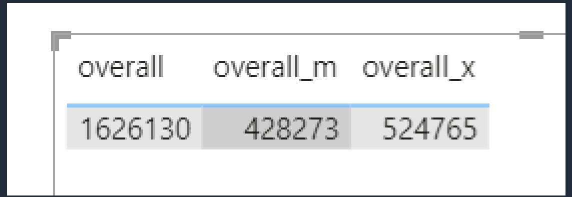
- take your calculated column DAX and make a measure using exactly the same code
  - `overall_m = SUM(ae_activity[over4])`
    - same code as the calculated column
    - different filter context
- then put `overall` and `overall_m` into a table
- then play with the filters, showing very different results for the calculated column and the measure:



KIND Learning Network

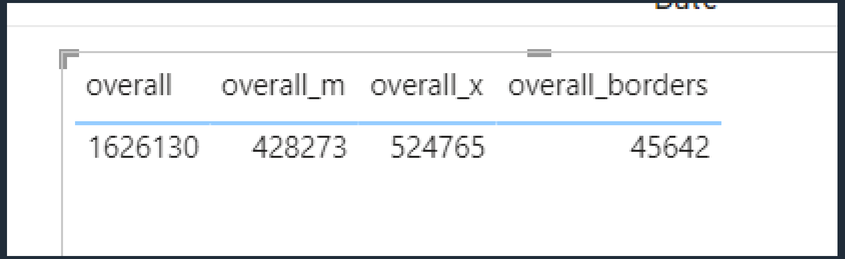
More on the filter context

- different functions interpret the context differently: SUM vs SUMX
  - SUMX evaluates some expression for each row in the context: `overall_x = SUMX(ae_activity, ae_activity[over4] + ae_activity[over8])`



overall	overall_m	overall_x
1626130	428273	524765

- CALCULATE as a function specifically for fooling with the filter context in a more detailed way
  - `overall_borders = CALCULATE(SUM(ae_activity[over4]), ae_activity[board] = "NHS Borders")` to restrict to just NHS Borders



overall	overall_m	overall_x	overall_borders
1626130	428273	524765	45642

KIND Learning Network

NHS Education for Scotland

Applications and best practice

- there's lots of overlap, and so you can work to suit your preferences
  - e.g. not clear whether creating calculated columns is better in DAX or PQM
- if you need your data to respond to the user, do it with DAX
- if you need to create lots of calculated values, do it with DAX
- if you need to transform your data, PQM
- if you need to clean and tidy your data, PQM
- if you need to undo/redo, PQM

Feedback and resources

- DAX: Russo and Ferrari 2019 *The Definitive Guide to DAX*
- PQM: Microsoft’s function reference is useful, but their intro pages are confusing and hard to recommend
- please can I ask for some feedback - takes less than a minute, completely anonymous, helps people like you find the right training for them

KIND Learning Network
NHS Education for Scotland