**1)**Create a new Power BI file, and load the **Voting figures** workbook from the above folder

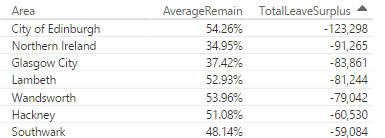
Create a measure to show for any filter context the total remain vote divided by the total electorate.  Here's what this should show:



*The first few rows of a table showing our measure by area (note that we've formatted it as a percentage).*

*The measure should be a simple calculation of the form****SUM(X) / SUM(Y)****.*

Too easy?  Create another measure to show the total value for any filter context of the difference between the leave and remain votes, to get something like this:

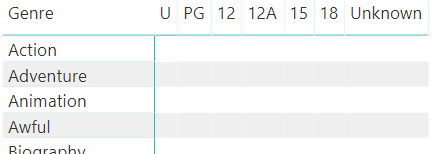


*Although it was only the 5th highest in the remain votes table by proportion of electorate,****City of Edinburgh****contributed by far the most votes to the remain cause.*

*Use the****SUMX****function to sum the expression****X - Y****over the****Voting****table, where****X****= the number of leave votes and****Y****= the number of remain votes.*

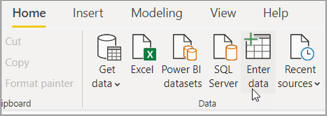
Save this as **Water under the bridge**, then close down the Power BI instance you're using.

**2)** Open the **films** file in Power BI Desktop.  It's missing a certain *je ne sais quoi*:



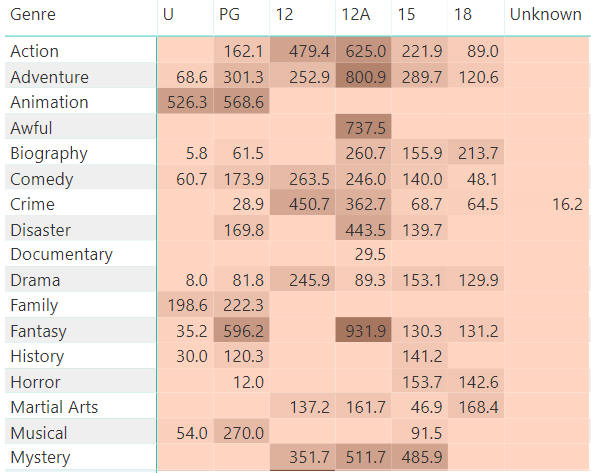
*Measureless despair!*

Create a table called **Film measures**to hold your measures:



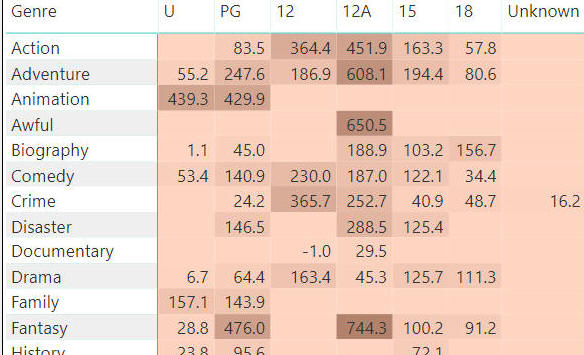
*As a reminder, in Power BI you can click on this tool to start creating a new table.*

Create a measure called **Average Box Office** to show the average box office takings (dividing by 1,000,000 to make the figures more readable):



*You can right-click on the measure in the field well to apply conditional formatting by this measure (in this case to show that 12A****Fantasy****films have the highest average box office takings - although if you scroll down you'll find 12 certificate****Romance****films do even better).*

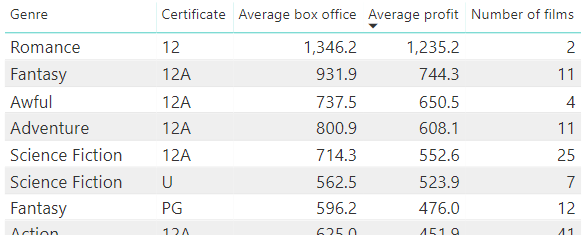
Create another measure called **Average Profit**, to show the average difference between the box office takings and budget for each film:



*Once more, 12A****Fantasy****films rule (although again, scrolling will reveal that 12 certificate****Romance****films do even better).*

*Because you're averaging an expression, you won't be able to use the standard****AVERAGE****function.*

Create one more measure called **Number of films**to count the number of films for the filter context (use **COUNTROWS**).  Create a table on a separate page to show your measures:

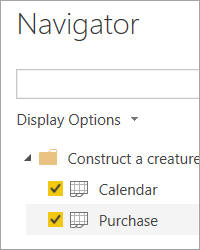


*Inside information: depressingly, the two films which propel 12 certificate romantic films to the top of the list are****Ghost****and****Titanic****.*

*You can click on the****Average profit****column heading to sort the genres in descending average profit order (although you probably knew that already ...).*

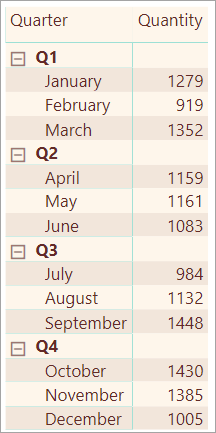
Save this as **Just let go**, then close it down.

**3)** Create a new Power BI file, and load the following tables from **Construct a creature including calendar**:



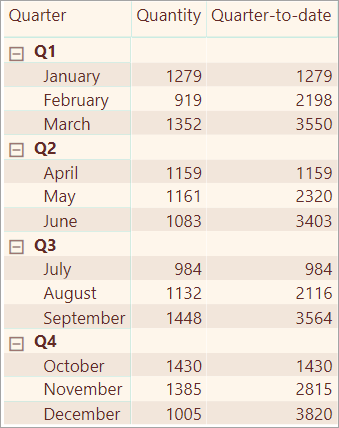
*You'll need to load both worksheets to do this exercise.*

 Create a matrix to show total sales by quarter and month for 2019, reading the notes below carefully!



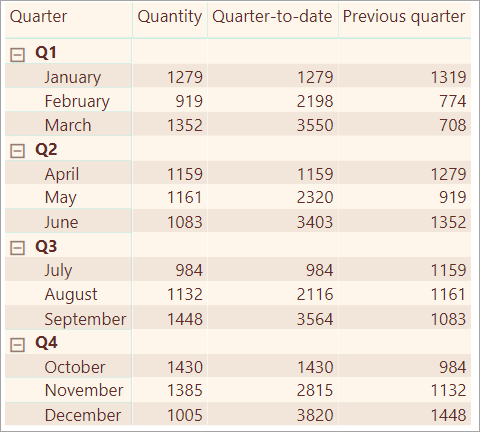
*You'll need to set up your calendar table correctly, create a relationship between the purchase date in the****Purchase****table and the calendar date in the****Calendar****table, and also apply a filter to the visualisation to show only figures for the calendar year 2019.*

Create and show a measure to show quarter-to-date figures:



*You can either use the****TOTALQTD****function, or the****CALCULATE****function (restricting the dates used with the****DATESQTD****function).*

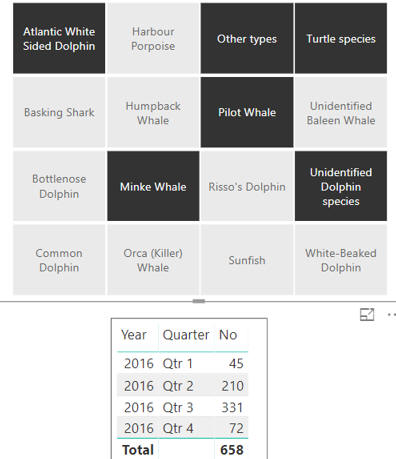
Now add another measure which shows for each time period what the sales were in the previous quarter:



*Use the****CALCULATE****function to sum quantity, but using the****DATEADD****function to go back one quarter in time from the filter context's dates.*

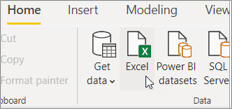
Save this file as **Tinder**, then close down the Power BI instance you're using (if you're wondering about the file name, it generates good dates if used properly ...).

**4)** Open the **Porpoiseful data** Power BI file in the above folder:



*The file contains sightings of certain marine animals in 2015-16.  You can choose species in the slicer and see a summary of sightings, but we want to be able to analyse the sightings in more detail.*

Load the calendar from the same folder into your model:



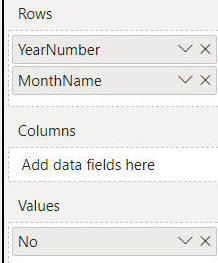
*The Power BI option to load data from Excel.*

To make this work:

1. Create a relationship between the calendar date key and the sighting Date column.
2. Make the calendar table a date table.
3. Choose to sort the calendar months by the MonthNumber column.

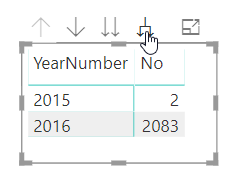
*Look on the Modeling of the Power BI ribbon tab for the last two options.*

Now create a matrix summarising sightings by year and month:



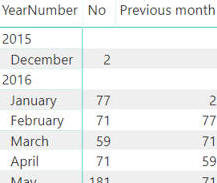
*The field well for your matrix should look like this. Be careful to tell Power BI not to sum the years, otherwise it will just add them together!*

You can now expand this to show the months too:



*Click on this tool to show the months too.  The figures are for Harbour Porpoise sightings.*

Use the PREVIOUSMONTH function to create a measure called Previous month to show sightings for the previous month:



*This measure displays total Harbour porpoise sightings in the previous month.*

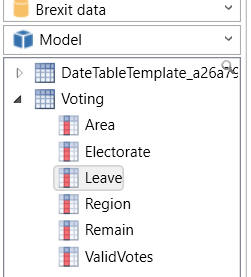
Now use the DATEADD function to create a measure called Two months ago which shows sightings two months previously:



*We're still looking at Harbour porpoise sightings.  Not sure whether this reveals anything, apart from the fact that you have got the hang of time intelligence functions in DAX!*

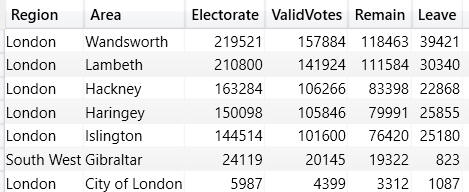
Save this file as Marine trends, then close it down.

**5)** Open the **Brexit data** Power BI Desktop file in the above folder, and connect to this file within DAX Studio:



*You should see something like this for the model's metadata.*

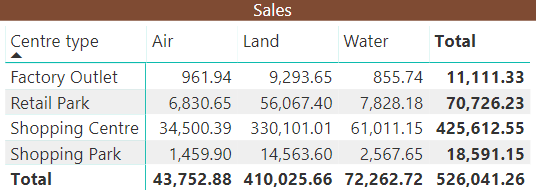
Write a DAX query using EVALUATE which lists out (in descending order of remain votes) data for those areas where the remain vote was more than three times the leave vote:



*What your query should show.*

Save this query as Remain strongholds.dax, then close it down.

**6)** Open **The Matrix** Power BI file in the above folder.  It shows the following matrix:

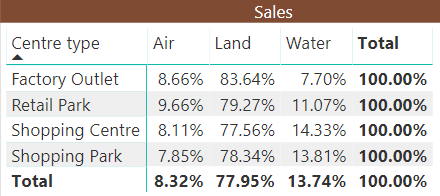


*Sales by environment and centre type*

Your task is to use the **CALCULATE**function in an expression to return the following measure:

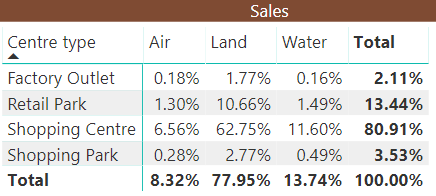
(Total sales) / (Total sales, but with the **Environment** constraint removed)

With this in mind, create a new measure in the **All measures**table called **All environment contribution**, and change your matrix to display this:



*Reassuringly, your environmental contributions should sum to 100% for each row.*

Create and display another measure called **Grand total contribution**which shows the ratio of sales to sales for all environments and all centre types:

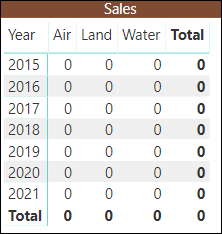


*Again reassuringly your grand total should be 100%.*

*You need to use the****ALL****function twice, to remove the environment constraint and the centre type constraint.*

Save this file as **No constraints**, then close it down.

**7)** Open **Woozle purchases** Power BI file in the above folder:



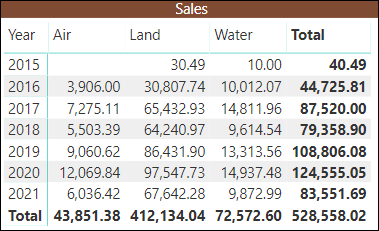
*At the moment the matrix shows a dummy measure.*

The **SUMX** function uses this syntax:



*The table can be a table, or the set of rows returned from a****FILTER****function.  We'll create examples of both types of syntax!*

Use this information to create a measure called **Sales**, which should sum the price multiplied by the quantity.  Show this measure in your matrix to give:

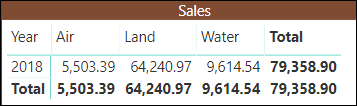


*You should see a report showing total sales by purchase year and environment.*

Copy the formula for this measure, and use this as a basis to create a new measure called **Sales in 2018**.  This measure should use the **FILTER**function to only sum sales where the year for the purchase date is 2018.

*You can use the****YEAR****function to return the year corresponding to any date.*

Show this measure in the matrix to get the following;

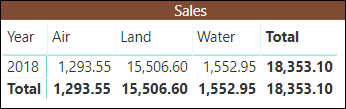


*The matrix collapses, as there are only figures for 2018.*

Now copy the formula for this measure again and amend it to create another measure called **Big sales 2018**, which only sums sales where:

* The purchase year is 2018; and
* The quantity of goods bought for a purchase was 5 or more.

This should give the following table:

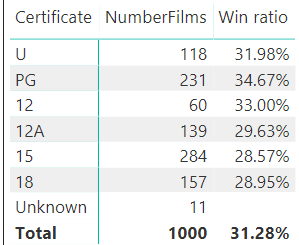


*The figures are falling, as we filter out more purchases.*

*You can use the****AND****function, the****&&****operator or nested****FILTER****functions to achieve this - the choice (as Graham used to say) is yours!*

Save this file as **Special measures**, then close it down.

**8)** Open the **Films** Power BI file in the above folder.  The aim of this exercise is to show the Oscar win ratio:



*The win ratio is the total number of Oscars won for the filter context divided by the total number of nominations.*

See below for how to proceed!

Create a measure called **Win ratio**, and within this declare two variables:

|  |  |
| --- | --- |
| **Variable** | **What it should hold** |
| **Nominations** | The total sum of all nominations for the filter context |
| **Wins** | The total sum of all wins for the filter context |

Your measure should then return the value of the second variable divided by the first (you could have done this without the variables, but they make everything clearer).

*You'll need to be careful to avoid showing a divide-by-zero error for the****Unknown****certificate category, for which there were no nominations.*

If you get this working, you could try commenting out different lines of your formula and returning different values to show, in turn:

* the number of nominations; and
* the number of wins.

Save this as **I love variables**, then close it down.

**9)** Open the **GBBO** Power BI desktop file in the above folder to see:



*Your task will be to divide the series into viewing figure bands.*

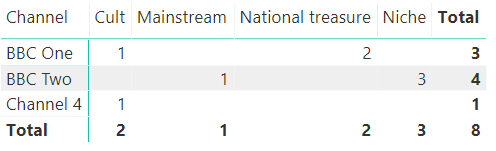
In the underlying table, create a new column called **Viewing Band**which uses the **SWITCH** function to allocate series to different bands according to these rules:

|  |  |
| --- | --- |
| **Million viewers** | **Band** |
| Up to and including 5 | **Niche** |
| 6 to 8 | **Mainstream** |
| 9 to 12 | **Cult** |
| More than 12 | **National treasure** |

The syntax of the **SWITCH** function is usually:

=SWITCH ( TRUE(), First condition, what if it's true, ... , Last condition, what if it's true, What to do if none of the conditions was true )

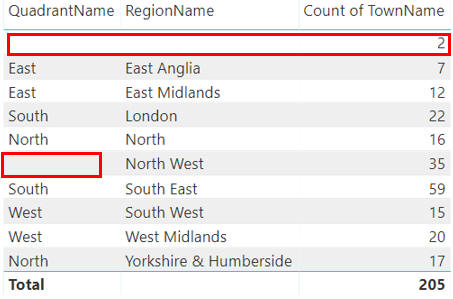
Turn the report's table into a matrix and use it to count how many series there were in each band for each channel:



*Channel 4 aren't national treasure level yet ...*

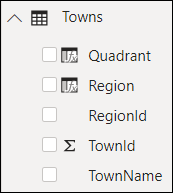
Save this file as **There was only one Rahul**, then close it down.

**10)** Open the **Town etc** Power BI report in the above folder.  It contains a table showing the quadrant, region and name for each town in the imported data:



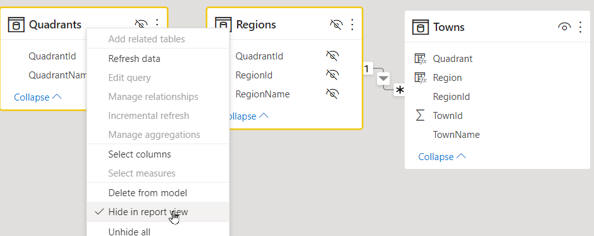
*However, two rows contain blanks, possibly because the id numbers don't match up between the tables.*

The aim of this exercise is to use the **RELATED** function to show the town and region within the quadrant table:



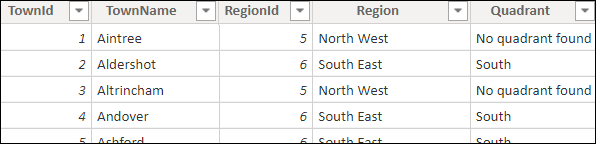
*The only visible table will be****Towns****, and the aim is to get it to show two calculated columns (unimaginatively called****Quadrant****and****Region****) to show the quadrant and region for each town. If you're feeling ambitious, you could try doing the exercise without reading any further instructions!*

To get this to work, first hide all of the tables apart from the town one:



*You can hide a table in****Model****view by right-clicking on it.*

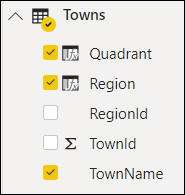
Now add two calculated columns to the towns table:



*The region column should show****No region assigned****if there is no related region in the regions table; the quadrant column should show****No quadrant found****similarly if there's no quadrant for a town.*

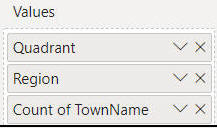
*Use the****RELATED****function to pick up columns from other tables, and the****ISBLANK****function to test if anything is returned.*

You should now see just one table in your **FIELDS** section:



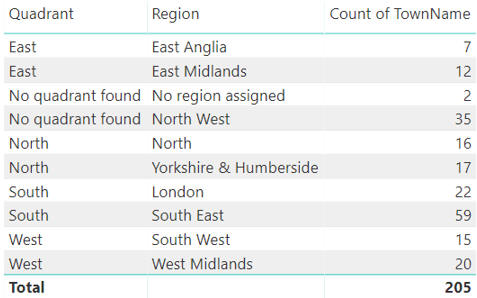
*The****Towns****table now contains for each town the corresponding region and quadrant.*

Change the fields displayed for your visual to get the required table:



*You'll need to display the quadrant and region fields from the****Town****table, rather than from the****Quadrant****or****Region****tables.*

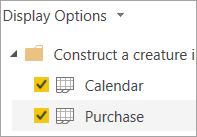
This should allow you to show a list of towns with no spaces in:



*Each town shows its region and quadrant, or****No region assigned****or****No quadrant found****as appropriate.*

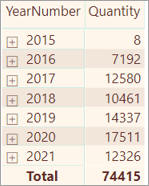
Save your revised file as **Where is Myton**, then close it down.

**11)** Create a new Power BI file, and load both tables from the workbook in the above folder:



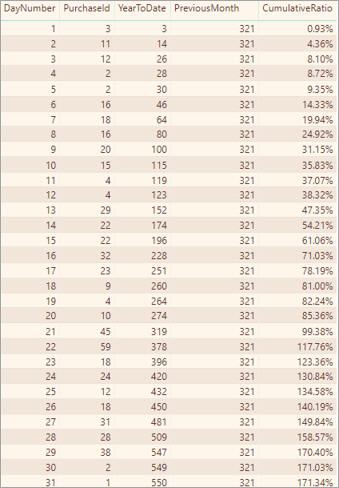
*The tables you'll need to load to do this exercise.*

Create a relationship between the two tables by the date field in each, and set your calendar table up to work correctly.  You should now be able to produce a matrix like this:



*You don't actually need to create this for the exercise, but it might give you the confidence you've set everything up correctly!*

The ultimate aim of this exercise is to create the following table:



*See below for lots of help on how to create these measures!  Note that you should filter your table to show only data for January 2019, otherwise you won't see these figures.*

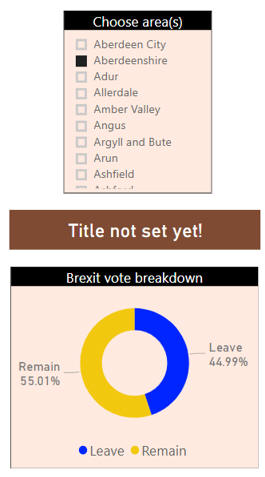
To do this, you'll need to create measures for the last 3 columns, using the guidance notes below:

|  |  |
| --- | --- |
| **Measure** | **Notes** |
| **YearToDate** | Count the cumulative year-to-date number of purchases, using either the **TOTALYTD** function or the **CALCULATE** and **DATESYTD** functions combined. |
| **PreviousMonth** | Use the **CALCULATE** function to count the number of purchases, with the **PARALLELPERIOD**function to change the filter context to the dates for the previous month. |
| **CumulativeRatio** | Divide the first measure by the second! |

*Remember to filter your table to show only figures for January 2019.*

Save this file as **Last month**, then close down the Power BI instance that you're using.

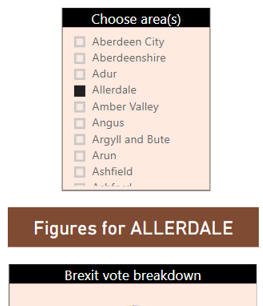
**12)** Open the **Vote Breakdown** Power BI file in the above folder, which should contain this report page:



*Choose an area at the top, and you'll see a breakdown of voting figures in the 2016 Brexit referendum (this is genuine data).*

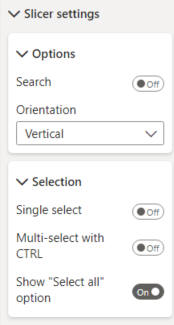
The title is based on a measure called **Title** - your task is to show the area chosen!

Edit the formula for the **Title** measure so that it displays the name of the area chosen, or **All areas**if you're not filtering by the **Voting[Area]**column:



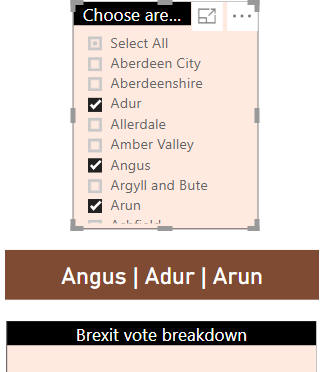
*Use the****ISFILTERED****function to see if you're filtering by the area, and the****VALUES****function to pick out the area name.*

If you have time, change the properties of your slicer to allow a user to select multiple areas and all areas:



*Choose these options for your slicer.*

Improve your **Title**measure so that it uses the **CONCATENATEX** function to show a list of areas chosen if a user ticks more than one:



*An example: this person has ticked 3 areas, so their names are listed in the title card.*

Save this as **Happy memories**, and close the file down.

**13)** Open the **Happy families** workbook in the above folder.  The aim of this exercise is to complete the table which currently has an empty column, so that it looks like this:



*The****Product list****column shows either****Just XXX****(if there's only one product in the filter context),****XXX, YYY, ZZZ****(if multiple products) or****Too many products to list****(if it's a total row).*

*In case you're feeling a creeping sensation of panic take hold of you, turn the page for step-by-step help!*

The first thing to do is to create a measure called **Product list**to cope with the main body of the table, and display this:



*We'll sort out the Total row in a bit ...*

To do this, create a measure which uses this logic:

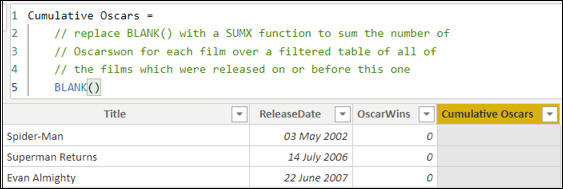
If there's only one value for the **ProductName**, concatenate the word **Just** with the product name; otherwise, concatenate all of the product names.

*Don't forget that you'll need to use the****VALUES****function to return a table of all of the product names for the filter context.*

Now add an additional **IF**function to your measure, and use the **ISFILTERED** function to show the text **Too many products to list**if you're filtering by the family name.

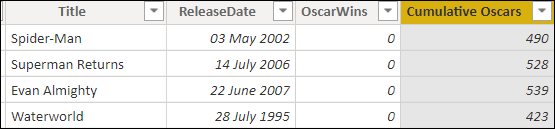
Save this file as **Happier families**, then close it down.

**14)** Open the **films1** Power BI file in the above folder.  Sadly, the running total column is blank:



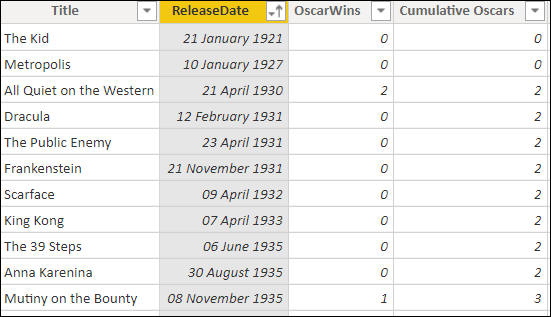
*Your task is to get the column to show what it says it should!*

Combine the **FILTER** and **EARLIER** functions to get the **Cumulative Oscars** column to be correct:



*The first few rows showing the cumulative total of Oscars won.*

The table makes much more sense if you sort it in release date order:

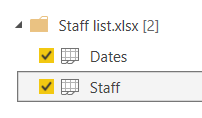


*Better - you can infer from this that****All Quiet on the Western Front****won 2 Oscars.*

Now create another calculated column called **Cumulative Oscars using variables** to do the same thing, but this time storing each film's release date in a variable first (and thus avoiding using the **EARLIER**function).

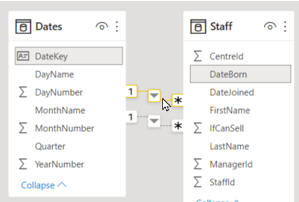
Save this file as **Oscar inflation**, then close it down.

**15)** Create a new Power BI Desktop file, and load both worksheets from the **Staff list** workbook in the above folder:



*One worksheet contains a list of staff with birth and joining dates; the other a list of dates.*

Make your calendar table a date table (it's always good practice!), and create two relationships:

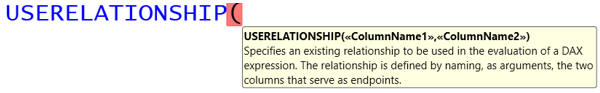


*One of the relationships joins the date key to the date someone was born; the other joins the date key to the date someone joined.*

 Create two measures:

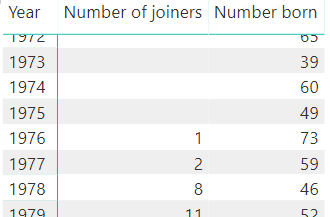
|  |  |
| --- | --- |
| Measure | What it should do |
| Number of joiners | Use the relationship to the DateJoined field to count how many staff members joined in a given time period. |
| Number born | Use the relationship to the DateBorn field to count how many staff members were born in a given time period. |

Here's the syntax of the USERELATIONHIP function as an *aide-memoire*:



*You need to specify the two columns at either end of your relationship.*

Create a table showing the value of each measure for each year:



*A small part of the table, which you can get to by scrolling down.*

 Save this as **Staff history**, then close it down.

**16)** Create a new Power BI Desktop file, and load the tables from the **construst-a-creature** workbook in the above folder into your model.  Using variables, create a measure which allows you to show this matrix:



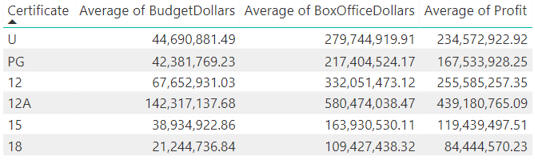
*The matrix shows for each cell the ratio between sales for that cell and sales for four- and six-legged animals for the same filter context.*

Here's a suggestion of variables to create within your measure:

|  |  |
| --- | --- |
| **Variable** | **What it could equal** |
| **FourLeggedSales** | The total quantity of sales for the current filter context, but for animals with 4 legs |
| **SixLeggedSales** | The total quantity of sales for the current filter context, but for animals with 6 legs |
| **ManyLeggedRatio** | A/B, where A = the total quantity of sales for the current filter context, and B = the sum of the previous two variables |

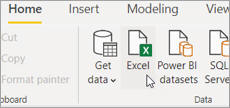
Save this workbook as **Variable fun**, then close down the Power BI instance you're using.

**17)** Open the **Films2** Power BI file in the above folder - you should see this table:



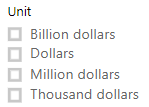
*Average box office takings, budget and profit by film certificate.*

Use the following menu option to import the **Units**worksheet from the above folder:



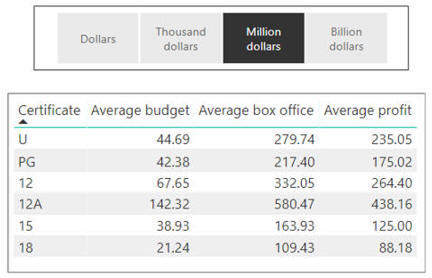
*Choose to import data from Excel.*

Create a slicer allowing you to choose which unit you want to use:



*A basic slicer - feel free to pimp yours, should you so wish.*

Use the **HASONEVALUE**function (among others!) to create 3 measures showing the average box office takings, budget and profit for the unit chosen:



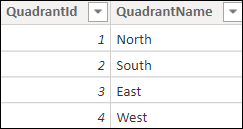
*Here we've chosen $m as our unit, so all of the figures are divided by 1000000.  If you don't choose a unit in your slicer your measures should all return blank, since you don't know what divisor to use.*

Still got time?  Add a measure called **Title**(and a card to display it) to show different titles according to whether or not you have chosen a unit:

|  |  |
| --- | --- |
| *Unit chosen* | *No unit chosen* |
| *Unit chosen* | *No unit chosen* |

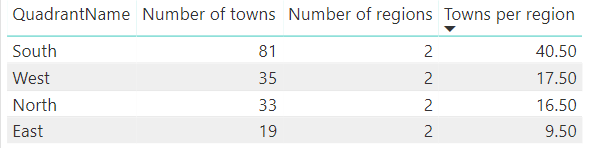
Save this file as **Whew**, then close it down.

**18)** Open the**Town** Power BI file in the above folder, and go to the quadrant table:



*This table contains the four quadrants of England!*

The aim of this exercise is to add 3 new columns to this table, so that you can show this visual:



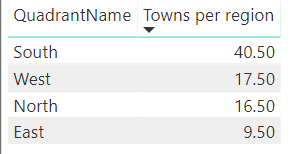
*The aim is to show for each quadrant how many regions it contains, how many towns it contains and what the ratio between the two figures is.*

Start by adding two new calculated columns to this table:

|  |  |
| --- | --- |
| **Column** | **What it should contain** |
| **Number of regions** | The number of regions in the related regions table. |
| **Number of towns** | The number of towns in the related towns table. |

*Use the****COUNTROWS****function to count how many rows there are in each related table.*

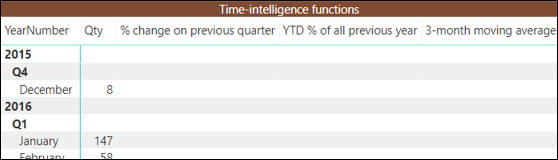
Now create a third calculated column which tells you how many towns there are in each region, and use this to show a table of the quadrants in town density order:



*Unsurprisingly, the South has the most towns in it (at least in this database).*

Save this as **Go East young woman**, then close it down.

**19)** Open the **incomplete report** Power BI report in the above folder, and see how many of the gaps you can fill in, given the time you have available!

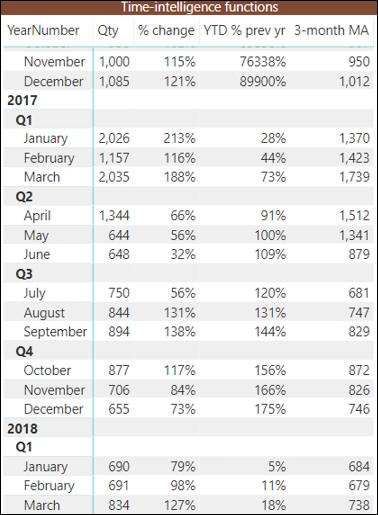


*Create 3 measures using the table below - the screen shot below shows what these should show.*

Here are the measures you need to create:

|  |  |  |
| --- | --- | --- |
| **No.** | **Measure** | **Thoughts** |
| 1 | **% of previous quarter** | For each quarter, what percentage the total quantity sold constitutes of the same figure for the previous quarter. |
| 2 | **YTD % of all previous year** | For each time period, the cumulative year-to-date figure divided by total quantity sold for the whole of the previous year. |
| 3 | **3-month moving average** | The average of the total quantity sold for the previous 3 months. |

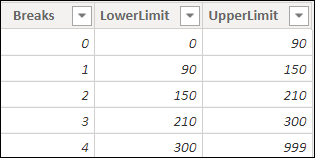
Here's what Wise Owl think the answers should look like!



*The start of the table, with the 3 measures filled in (referred to by abbreviated names to save space).*

Save your file as **Desperate measures**, then close it down.

**20)** Open the **list films** Power BI file in the above folder.  As well as a **list of films**, it contains a table giving how many times the author would need to pop out for a comfort break during each film's screening:



*From this table you should be able to make a reasonable guess at the author's age ...*

*Note that the two tables are not joined together - nor should they be for this exercise.*

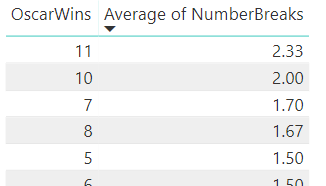
In the **Films**table create a calculated column called **NumberBreaks** which shows for each film the number of breaks needed.  To do this you want to pick out the value of the **Breaks** column where:

* The value of the lower limit in the **ComfortBreaks** table is less than or equal to this film's running time in minutes; and
* The value of the upper limit in the **ComfortBreaks** table is greater than this film's running time in minutes.

To refer to *this film*you can either use the **EARLIER** function or create a variable.

*Because you're not using an aggregation function, you'll need to use the****VALUES****function within****CALCULATE****to pick out what you can be confident is the only****Breaks****column value returned.*

When you've finished, you should be able to create the following matrix:

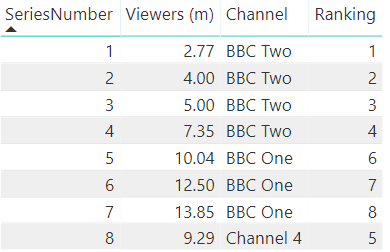


*You're going to be popping out more to see multiple-Oscar films like****Titanic****and****The Return of the King****than for less successful films, because they last longer on average.*

If you still have time, try creating a second calculated column to do the same thing, but this time using the method you didn't use the first time.

Save this file as **Wee report**, then close it down.

**21)** Open the **GBBO** Power BI file in the above folder, and create a measure (not a calculated column - that would be too easy) to rank each series by its viewing figures to get:



*The ranking more or less matches the series number, but there's a blip where GBBO went over to Channel 4.*

*This exercise may look deceptively easy, but it's probably the hardest one in the set!*

Save this as **Rank expression**, then close it down.