DA\_Lab\_Assignment –(Power BI ) –3

Refere the link to study Power BI <https://docs.microsoft.com/en-us/power-bi/desktop-get-the-desktop>.

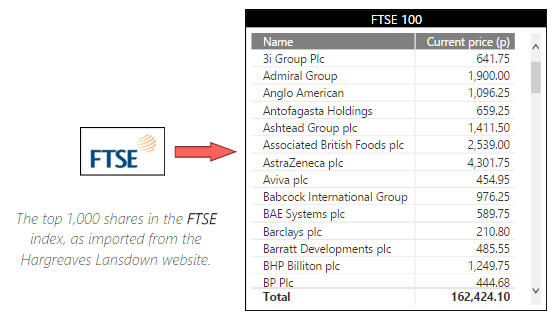
* Use <https://docs.microsoft.com/en-us/power-bi/desktop-get-the-desktop>  
  And Study following points from the link.  
  Common query tasks in Power BI Desktop  
  Create and manage relationships in Power BI Desktop  
  Data View in Power BI Desktop  
  Query overview in Power BI Desktop  
  DAX basics in Power BI Desktop  
  Relationship View in Power BI Desktop  
  Report View in Power BI Desktop  
  Connect to data in Power BI Desktop  
  Custom visuals in Power BI  
  Part I, Add visualizations to a Power BI report  
  Part 2, Add visualizations to a Power BI report  
  Customize visualization titles, legends, and backgrounds  
  Shape Maps in Power BI Desktop   
  Combine files (binaries) in Power BI Desktop  
  Aggregations in Power BI Desktop   
  Use composite models in Power BI Desktop (preview)  
  Many-to-many relationships in Power BI Desktop (preview)  
  Storage mode in Power BI Desktop   
  Using Direct Query in Power BI  
    
  A MCQ based on above points will be scheduled in the next lab session for 10 marks.  
  After Studying above topics Follow the steps to complete the PBI Assignment.  
  Use the following URL given below   
  <https://www.wiseowl.co.uk/power-bi/exercises/power-bi-desktop/>  
  1. Complete all 36 assignments on power BI.  
  2. Complete any one dashboard from sample dataset.  
  Deadline :30 November 2018.

1. Use the XLS file films

Create a new Power BI Desktop file, and get a list of the top 1000 films of all time, as rated by **They Shoot Pictures, Don't They?**

Save this Power BI file with the name **Citizen bloody Kane again**, then exit this instance of Power BI Desktop.

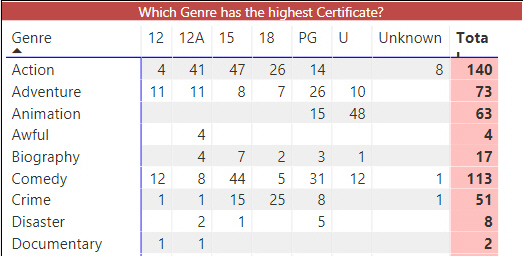
1. Create a new Power BI Desktop file, and get a list of the FTSE 100 shares Add the image in the above folder, a shape, a table and a text box to get a report looking something like this:



You can rotate the arrow by adding an arrow shape and changing its **Rotation** property.

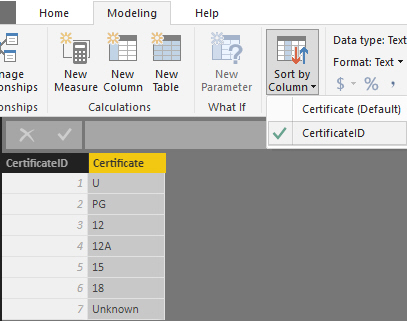
Save this Power BI file as **Footsie**, then exit this instance of Power BI Desktop.

1. Using the **Age specific Genres.xlsx** file in the above folder create a matrix showing the **Count** of films by **Certificate** rating and **Genre:**



Your matrix will have all the certificates listed.

Notice how the certificates are in alphabetical order? **CertificateID** is in a more sensible order; use this to reorder the **Certificate** column:



Sometimes the only way to get the order you want is to create a calculated column or separate table with a numbering system!

You now have a more familiar order:

Matrix Order Sort by column

Apparently the **Action** genre is too scary for young kids?

Lastly, to make the report a little more interesting add the images found in the above folder to the page like this:

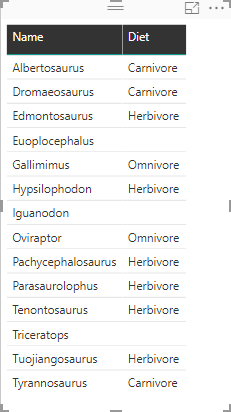


Feel free to use a more interesting background colour.

Optionally save this report as **Still a better matrix than reloaded.pbix** and close it down.

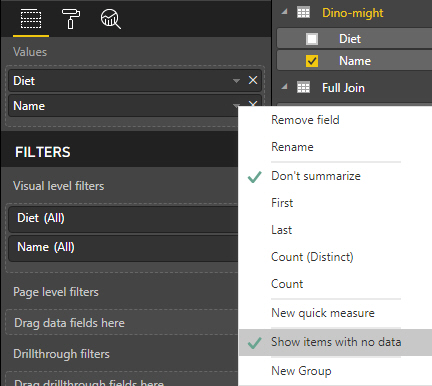
This page has [1 thread](https://www.wiseowl.co.uk/power-bi/exercises/power-bi-desktop/basic-reports/4368/#posts)

Make a table holding a list of dinosaurs and their preferred diet, which surprisingly isn't Atkins. **Dino Diets.xlsx** in the above folder will help!



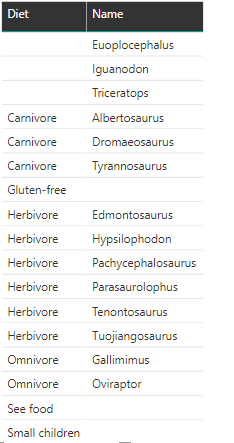
Some of these dinosaurs appear to eat the same as size 0 models (nothing)!

1. The reason some of these are empty is due to the **Dino.DietID** not having a corresponding **Food.DietID.** What about the missing **Diets**? Go to the field well:



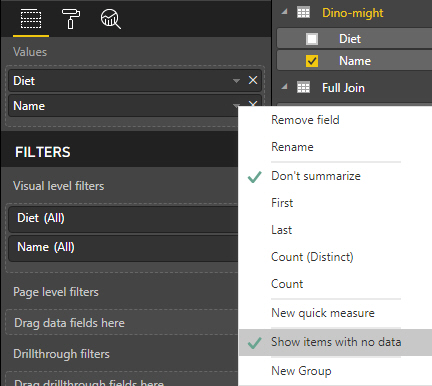
Ticking **Show items with no data** will unhide any missing food types.

Now in the table you should see all the values from both tables shown: dinos with an unknown diet, and diets with no dino to eat them:



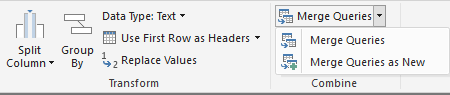
See food like Sea food? Eats all the food it sees, get it?

That was how to show all the results. What if we don't want blanks? Getting rid of **Diet** blanks is easy. Simply untick **Show items with no data:**



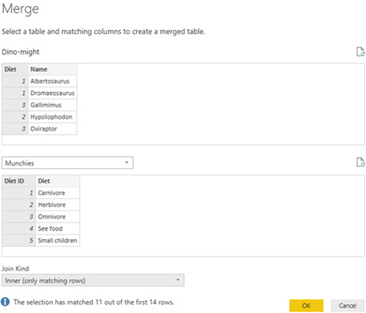
Sadly it isn't as straight forward for **Dino** as we can't untick anything.

Open the **Query editor** and choose **Merge Queries:**



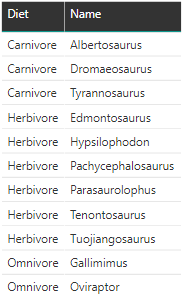
**Merge queries** will filter the table you have highlighted (**Dino**) whilst **Merge Queries as New** will make a new filtered table.

We want to remove all the dinosaurs who haven't got a listed food source  In this case we've chosen **Merge queries** to filter the existing table. Click the two **DietID** columns:



The default kind of join was **Left Outer Join**, which was returning all dinosaurs but only matching diets. Change the join type to **Inner**.

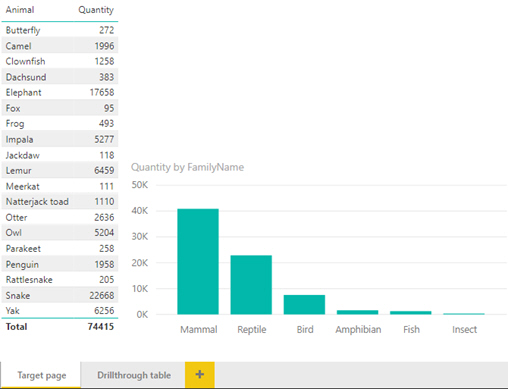
Using an inner join will mean that only dinosaurs with known diets will be shown. Similarly only diets eaten by a known dinosaur will be shown (we can assume the rest are extinct).



This does mean you can't change your mind however. Trying to use the show data option now will have no affect on which dinosaurs can be seen.

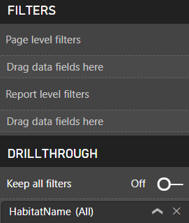
Optionally save this as **A diet to die for.pbix** and close it down.

5. Start by creating the following table and chart using the **Drilling down to details.xlsx** file found in the above folder (choose to import all of the tables listed).



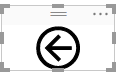
This is our main data page and the one we will filter through to, when someone uses drill-through on a habitat.

To make this into a page that will be filtered, drag the **HabitatName** into the **Drillthrough** **filters** section of the field well:



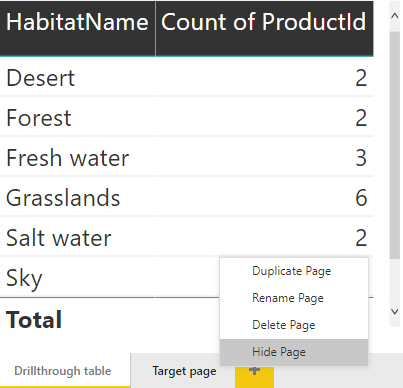
Whilst only one drill-through filter can be applied at once, you can have as many potential filter fields as you like.

Power BI will create a back button automatically after you add a field to **Drillthrough filters.** Only one button will be created no matter how many fields are added.



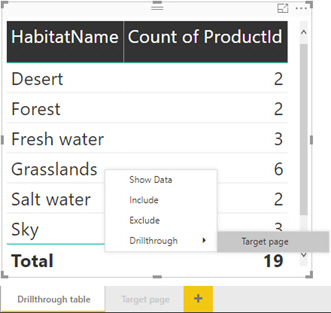
The back button is a really handy tool. It returns the user to the page they came from last!

Now to create the page to drill from. On a separate page called **Drillthrough table** create the following table:



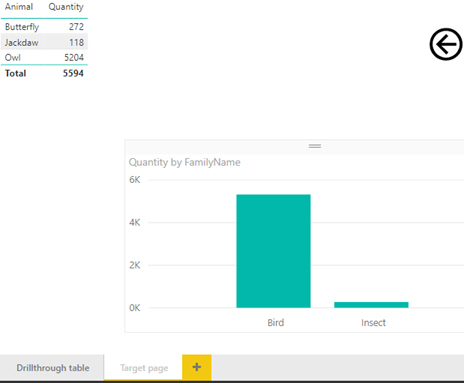
You could also at this point hide the page called **Target page** that you want to drill to. This prevents anyone getting to that page without using the drill-through.

Surprisingly that is all that you need to do. Now visuals that contain a field include in the **Drillthrough filters** can be right clicked, to navigate to the filtered page.



As you add more pages that have a drillt-hrough filter on them, more options will appear in the **Drillthrough** section shown.

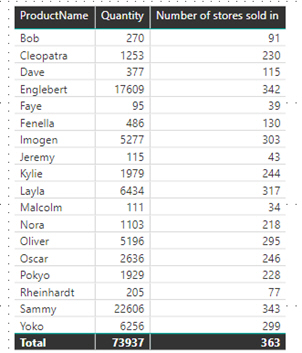
Finally the result of all your hard work should be that the hidden report page is automatically filtered:



So if I have 5,322 birds in the hand does that mean they are worth 10,644 in the bush?

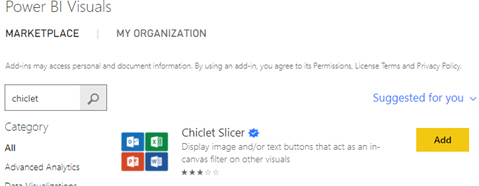
Optionally save this as **Who you calling a bird.pbix** and close it down.

6. Use the folder Chilclet slicer datafiles ,Create a basic table using the **Click the cutest critter** workbook in the above folder:



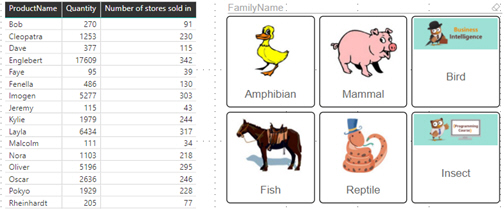
**CountDistinct** gives the number of stores each product was sold in.

Now create a slicer, but using images instead of just the values.  To do this add the **Chiclet Slicer** custom visual from the **Marketplace**:



If you don't have access to the internet you can find a copy of the visual in the above folder.

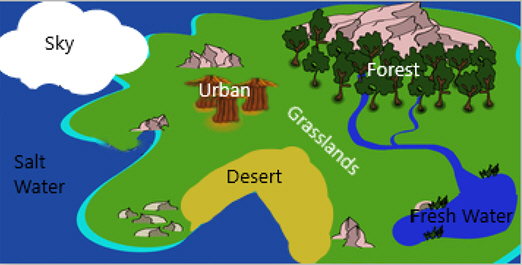
Make a new table which holds the URLs of appropriate images for each of the different values in the **FamilyName** column, or import data from the **Web links** Excel file:



It's a sea-horse, a programming BUG and am-phi-bin about the duck, sorry.

Optionally save this as **Shoddy slicer substitutes** and close it down.

7.. In the Treasure map folder is an image with several different habitats on it. Using the **Synoptic Panel** custom visual, map the data from the Excel file onto the map.

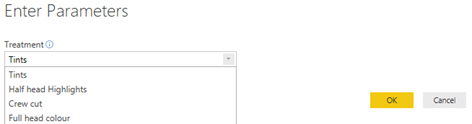


The easiest way to add areas to images is to visit https://synoptic.design/

Clicking on an area should filter other visuals on the page, so make sure it's paired properly!

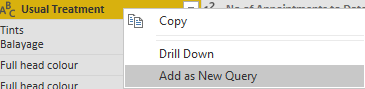
Optionally save this as **Treasure Map** then close it down.

8. Use Hair Saloon Dataset , Rather than give an input box for parameters we instead want to create a drop down for filtering. Like this in fact:



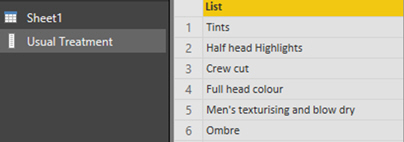
Choosing a treatment will filter the data brought through into the data model.

Load the data from the Excel sheet in the above folder. Before we create the parameter we need to create the drop down list. Right-click **Usual Treatment** column:



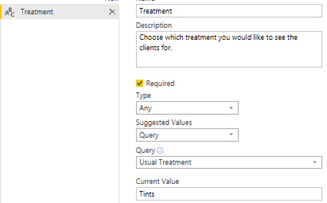
At the bottom of the list choose **Add as New Query.**

By creating a separate query based on the original query we ensure that the drop down values will be the same as those in the original column.



This list will contain each value multiple times, so it's worth right clicking and choosing **Remove Duplicates**.

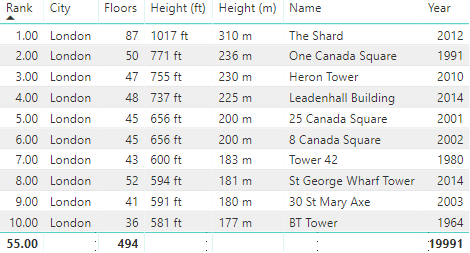
You now have a query to use for populating a drop down. Create your parameter:



Irritatingly you still must choose an initial **Current Value** (and even more annoyingly it doesn't even give you the drop down you made to help you choose!).

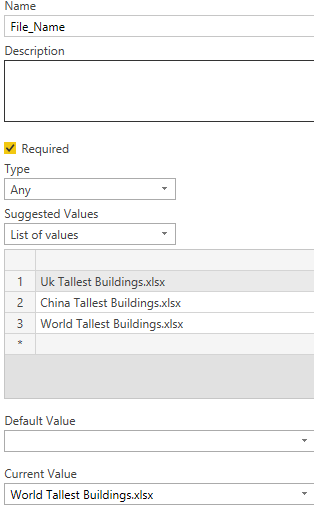
Try flicking between the different treatments using **Edit Queries** then **Edit Parameters** from the **Home** tab. Save as **Dropping usual treatments.**

9. In the tallest tower folder are 3 Excel files containing information about the tallest buildings in various countries. The aim is to be able to switch between viewing the top 10 from each file by switching which is connected:



To start with either connect to the UK file from scratch or else open the **PBIX** file in the above folder.

We need to store the location of each file in a parameter so we can switch between them. Create a parameter and list the file names:



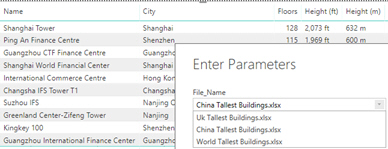
This will create a drop down letting users choose which file to connect to.

Now access the **Advanced Editor** from the **Home** tab within the **Query Editor** and replace the file name with your parameter:

Power BI Query Editor Dynamic Connection string

The **&** is used to concatenate the string connection with the value inside the parameter.

Exit **Query Editor** and then use **Edit Parameters** under **Edit Queries** to switch to a different file:



You may need to click **Apply changes** to finish loading the data.

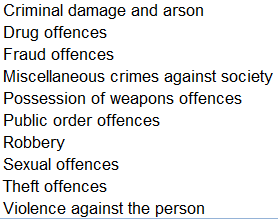
Optionally save this as **Tall, Taller, Tallest** and then close it down.

10 Use criminal dataset Create a new Power BI Desktop file, and load the Excel workbook from the folder above.  Use this to create a table showing the number of offences for each offence subgroup (you can see a link to the original data source [here](https://www.wiseowl.co.uk/sundry/pbd1/), if you're interested):



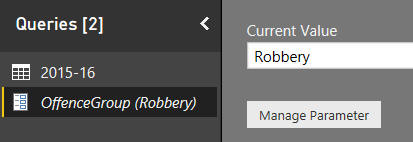
The first few roles of the table.

We want to be able to filter this to show only those offences for a particular group:



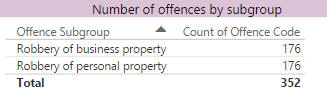
The offence groups. It would be nice to be able to type one of these into a parameter box, and see the table change!

In Query Editor, create a new parameter called **OffenceGroup**, and set its default value to **Robbery** (if for no other reason than that it's the shortest to type in!):



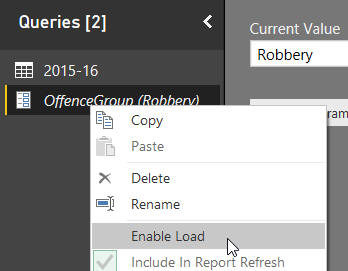
Your parameter should look something like this.

Still in Query Editor, filter the table of offences to show only those ones for the offence group specified by your parameter:



After returning to your report, you should now see your table is much smaller!

We could do with a title for our report.  Go back to Query Editor, and tell Power BI Desktop that you want to load your parameter into your data model:

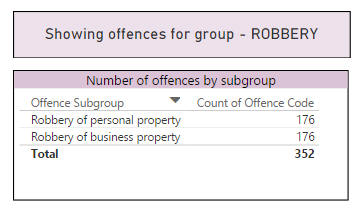


In Query Editor, choose to enable loading for your parameter.

Now add a measure to the main data table, with a formula looking something like this:

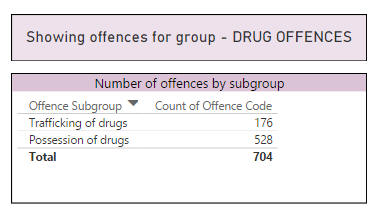
Title = "Showing offences for group - " &   
UPPER(values(OffenceGroup[OffenceGroup]))

Display this measure in a card to get a nice-looking report:



You'll need to turn the card's Category Labels off to get this effect.

Go into Query Editor and change the offence group to **Drug offences**, and check that your table then updates:



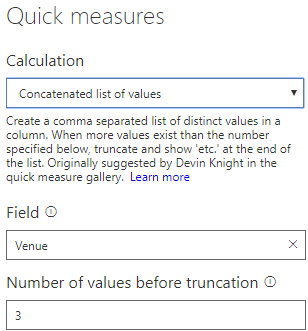
To avoid getting an error, you'll need to refresh your report to bring in the changed value of the parameter (the table will do this, but the card won't).

Save this as **Was it worth it**, then close down the Power BI instance you're using.

11. Use Dynamic Titles excel sheet, Using the Excel file in the above folder create a chart, a slicer and apply a page filter. The results should look something like this:

Wouldn't it be nice to have a message box saying which filters have been applied?

From the **Home** tab choose **Quick Measure**, then from the bottom of the list choose **Concatenated list of values:**



All the values in the **Field** column will be concatenated.  The **Number of values before truncation** boxindicates how many are listed before returning **etc**.

Repeat this for all the columns you have used for filtering, and then insert the newly created measures into cards:



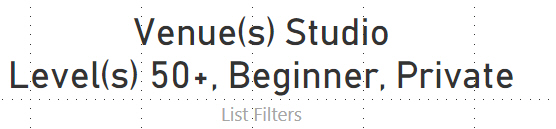
What would be slightly nicer is having them in one card together, but there isn't a quick measure to do this.

From the **Home** tab choose **New measure** and concatenate the two previously created measures together:

Power BI Measure

The **UNICHAR(10)** returns a line break, so the filters will be listed on two lines.

Put this in to a card to create your new dynamic title:



Try using this technique with **Parameters** and also with **Drill through** to make truly intuitive titles.

Optionally save this as **Helpful headlines** and close it down.