

## Лабораторная работа 15

### Part 1

Before you can do this exercise, you'll need to download and unzip [this file](#) (if you have any problems doing this, [click here](#) for help).

Movies come in a variety of different genres - these are currently stored in separate workbooks. Using the **Folder** data source option combine all the genres into a single data model table:

Content	Name	Extension
Binary	Action Films.xlsx	.xlsx
Binary	Awful Films.xxx	.xxx
Binary	Comedy Films.xlsx	.xlsx
Binary	Romance Films.xlsx	.xlsx

The files in the above folder that you want to combine (but see hint below).

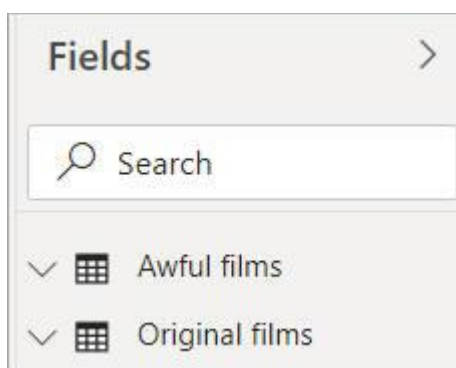
Note that one of the files above has an **.xxx** extension (and so won't import properly), so after choosing to combine your files be sure to tick the **Skip files with errors** box.

This will give you 321 films, in 3 different genres. Return to Power BI proper and create a **Table** visual to compare their lengths:

Genre	Average of RunTimeMinutes
Action	117.44
Comedy	106.25
Romance	130.68
<b>Total</b> . . . . .	<b>113.68</b> .

Your table should show the average run time for each genre.

It would appear that the shorter the movie, the better it is. Let's test this hypothesis by adding some truly terrible movies into our data model! Firstly, rename the file **Awful Films.xxx** in the above folder to **Awful Films.xlsx**, then load it into your data model:



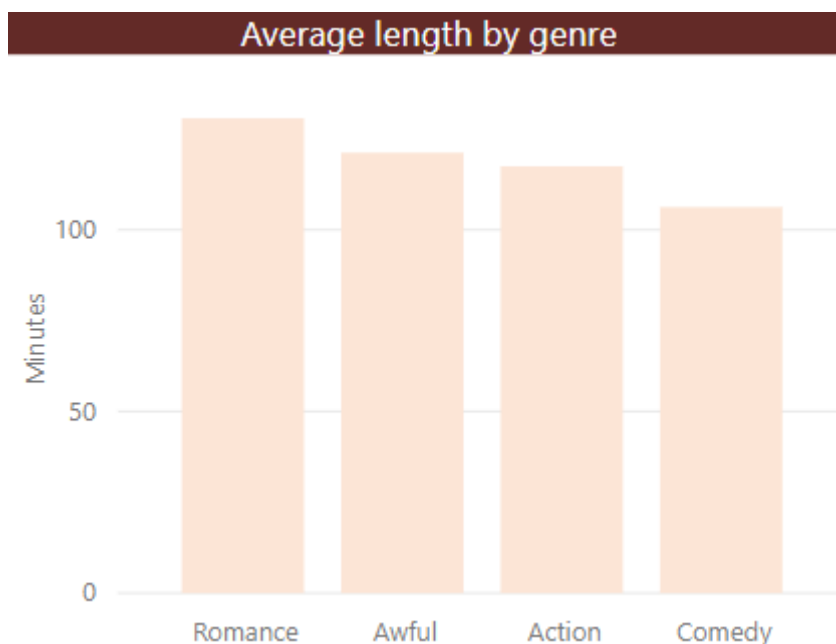
You now have two tables (which we've renamed), which you should combine as explained below.

Use the **Append** option in query editor to append one of these queries onto the other. You should now have 326 films in your combined table, not 321:

	<sup>12</sup> <sub>3</sub> FilmID	<sup>A</sup> <sub>C</sub> Genre	<sup>A</sup> <sub>C</sub> Title
1	295	Awful	The Twilight Saga: Breaking Dawn - Part 2
2	303	Awful	The Twilight Saga: Breaking Dawn - Part 1
3	315	Awful	The Twilight Saga: Eclipse
4	326	Awful	The Twilight Saga: New Moon
5	1574	Awful	Twilight
6	5	Action	Superman Returns

The combined table of films, beginning with the 5 awful ones (at least according to Wise Owl).

Create a column chart based upon the combined table to test your hypothesis:



It seems as if the hypothesis doesn't stand up in the face of data!

Optionally save this as **Are shorter movies better**, then close it down.

## Part 2

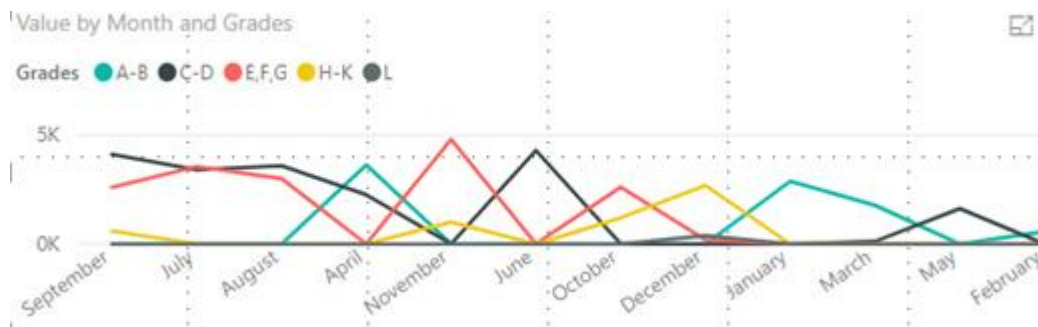
Before you can do this exercise, you'll need to download and unzip [this file](#) (if you have any problems doing this, [click here](#) for help).

One of your colleagues has created an excellent pivot table within Excel. While it may look excellent there, however, it causes problems within the data model:

	January	February	March	April	May	June
A-B	2887	570	1782	3633	0	0
C-D	0	0	117	2264	1631	4304
E,F,G	0	0	0	0	0	0
H-K	0	0	0	0	0	0
L	0	0	0	0	0	0

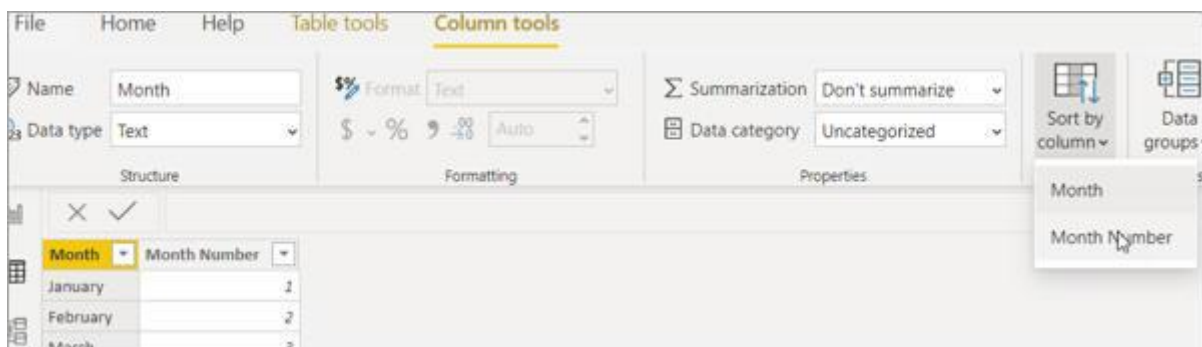
The first half of the year certainly seems to be having a promising start. Wonder if the second half is as good?

Bring this data into **Query Editor** and unpivot it so that it can be presented within report view. Create a line chart based on these results:



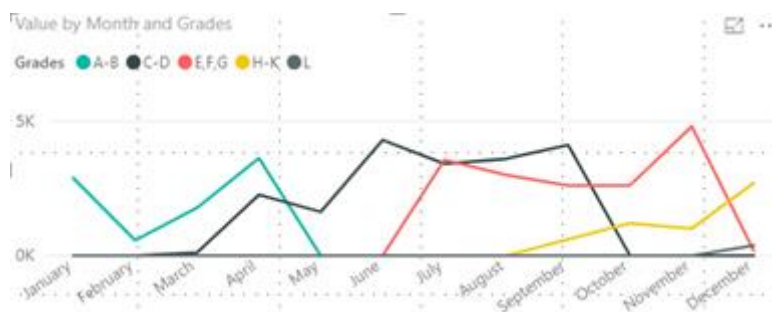
Yikes! We can't tell anything from this due to the weird month order.

Import the month numbers table from the Excel file in the above folder. Join this to your pivot table via the month names (using the month number to sort the month names):



A handy feature which makes our months appear in a more logical order.

Recreate the line chart using the sorted month column from the other table, admire your sorted months:



Ouch. Guess the summer holidays didn't help with focussing the mind!

Optionally save this as **More grades, more green** and then close it down.

### Part 3

Before you can do this exercise, you'll need to download and unzip [this file](#) (if you have any problems doing this, [click here](#) for help).

Some IT savvy palaeontologist has separated the fun facts about dinosaurs from their names and pronunciations, leaving us to guess which dinosaurs these are:

17654	Cretaceous	carnivore	9	3.5	1500
25126	Jurassic	carnivore	12	5	2000
13664	Cretaceous	carnivore	10	4	2000

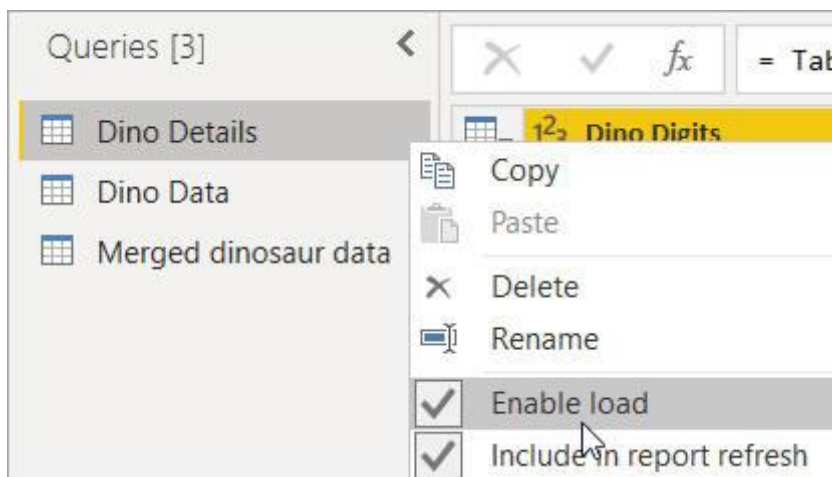
We don't know what to call it - but this dino probably calls you dinner!

Using the CSV and Excel files in the above folder, merge the data together into a new table in your data model. This should look like this:

Dino Digits	A <sup>B</sup> <sub>C</sub> Name	1.2 Length (m)	1.2 Height (m)	1.2 Mass kg	A <sup>B</sup> <sub>C</sub> Diet
17654	Albertosaurus	9	3.5	1500	carnivore
25126	Allosaurus	12	5	2000	carnivore
13664	Baryonyx	10	4	2000	carnivore

The other columns are still there just off the page (probably too scared to be near these beasts).

Since we have created a new table we don't want to import the duplicated data from our original files. Stop them from loading into the data model:



This lets us reference the original files without making our pbix file any larger with repeated data.

Optionally save this as **Digging for dinos**, then close it down.

### Part 4

Before you can do this exercise, you'll need to download and unzip [this file](#) (if you have any problems doing this, [click here](#) for help).

Open the **pbix** file called **Outdated Movies** from the above folder, and have a look at the table visual it contains (already created in report view):

Title (Rank)	Year of release	Rating
The Wizard of Oz (1)	1939	99.00%
The Third Man (2)	1949	100.00%
Citizen Kane (3)	1941	100.00%
Das Cabinet des Dr. Caligari. (The Cabinet of Dr. Caligari) (4)	1920	100.00%
All About Eve (5)	1950	100.00%
The Godfather (6)	1972	99.00%
Inside Out (7)	2015	98.00%
Modern Times (8)	1936	100.00%
Metropolis (9)	1927	99.00%
E.T. The Extra-Terrestrial (10)	1982	98.00%
<b>Total</b>		<b>993.00%</b>

The top 10 movies according to **Rotten Tomatoes** as of 2016.

What we want to do is update this list to show the current top 10 films on the website. The easiest way to do this is to point the M language instructions at a different data source. First connect to the **Rotten Tomatoes** website:

From Web

Basic ☒ Advanced ☐

URL

OK Cancel

Since this is a website you may find the answer needs a refresh! As of writing the table we want is **Table 2**.

Once you have the data, go to the original top 100 table and from the **Home** tab choose **Advanced editor**, sadly the old connection won't work but we still get the code:

```
let
Source = Csv.Document(File.Contents("C:\Users\sam.lourie\Desktop\Power BI Desktop\files needed\Top100Films.csv"),[Delimiter=";", Columns=4,
#"Promoted Headers" = Table.PromoteHeaders(Source, [PromoteAllScalars=true]),
#"Changed Type" = Table.TransformColumnTypes(#"Promoted Headers",{{"RatingTomatoMeter", Percentage.Type}, {"Title", type text}, {"No. of R
#"Reordered Columns" = Table.ReorderColumns(#"Changed Type",{"Title", "No. of Reviews", "Rank", "RatingTomatoMeter"}),
#"Renamed Columns" = Table.RenameColumns(#"Reordered Columns",{{"RatingTomatoMeter", "Rating"}),
#"Move rank column" = Table.ReorderColumns(#"Renamed Columns",{"Rank", "Title", "No. of Reviews", "Rating"}),
#"Removed Columns" = Table.RemoveColumns(#"Move rank column",{"No. of Reviews"}),
#"Sorted Rows" = Table.Sort(#"Removed Columns",{{"Rank", Order.Ascending}}),
#"Filtered Rows" = Table.SelectRows(#"Sorted Rows", each [Rank] <= 10),
#"Split Column by Delimiter" = Table.SplitColumn(#"Filtered Rows", "Title", Splitter.SplitTextByEachDelimiter({"("}, QuoteStyle.Csv, true),
#"Replaced Value" = Table.ReplaceValue(#"Split Column by Delimiter", "(", " ", Replacer.ReplaceText, {"Title.2"}),
#"Changed Type1" = Table.TransformColumnTypes(#"Replaced Value",{{"Title.1", type text}, {"Title.2", type text}}),
#"Renamed Columns1" = Table.RenameColumns(#"Changed Type1",{{"Title.2", "Year of release"}),
#"Inserted Merged Column" = Table.AddColumn(#"Renamed Columns1", "Merged", each Text.Combine({Title.1, "(", Text.From([Rank], "en-GB"}),
#"Renamed Columns2" = Table.RenameColumns(#"Inserted Merged Column",{{"Merged", "Title (Rank)"}),
#"Reordered Columns1" = Table.ReorderColumns(#"Renamed Columns2",{"Title.1", "Rank", "Year of release", "Rating", "Title (Rank)"}),
#"Removed Columns1" = Table.RemoveColumns(#"Reordered Columns1",{"Title.1", "Rank"}),
#"Reordered Columns2" = Table.ReorderColumns(#"Removed Columns1",{"Title (Rank)", "Year of release", "Rating"}),
#"Reordered Columns2"
```

Copy all the steps from '**Reordered columns**' down. We don't want a CSV file - instead we want our web connection!

Open up the **Advanced editor** for your web connection to the **Rotten Tomatoes** website:

```
let
Source = Web.Page(Web.Contents("https://www.rottentomatoes.com/top/bestofrt/")),
Data2 = Source(2)[Data],
#Changed Type = Table.TransformColumnTypes(Data2,{{"Rank", Int64.Type}, {"RatingTomatometer", Percentage.Type}, {"Title", type text}}, {"Title", "Rank", "RatingTomatometer"}),
#Reordered Columns = Table.ReorderColumns(#Changed Type,{"Title", "No. of Reviews", "Rank", "RatingTomatometer"}),
#Renamed Columns = Table.RenameColumns(#Reordered Columns,{{"RatingTomatometer", "Rating"}}),
#Move rank column = Table.ReorderColumns(#Renamed Columns,{"Rank", "Title", "No. of Reviews", "Rating"}),
#Removed Columns = Table.RemoveColumns(#Move rank column,{"No. of Reviews"}),
#Sorted Rows = Table.Sort(#Removed Columns,{{"Rank", Order.Ascending}}),
#Filtered Rows = Table.SelectRows(#Sorted Rows, each [Rank] <= 10),
#Split Column by Delimiter = Table.SplitColumn(#Filtered Rows, "Title", Splitter.SplitTextByEachDelimiter({"("}), QuoteStyle.Csv, true),
#Replaced Value = Table.ReplaceValue(#Split Column by Delimiter,"(", "", Replacer.ReplaceText,{"Title.2"}),
#Changed Type1 = Table.TransformColumnTypes(#Replaced Value,{{"Title.1", type text}, {"Title.2", type text}}),
#Renamed Columns1 = Table.RenameColumns(#Changed Type1,{{"Title.2", "Year of release"}}),
#Inserted Merged Column = Table.AddColumn(#Renamed Columns1, "Merged", each Text.Combine({[Title.1], "-", Text.From([Rank], "en-GB")}),
#Renamed Columns2 = Table.RenameColumns(#Inserted Merged Column,{{"Merged", "Title (Rank)"}}),
#Reordered Columns1 = Table.ReorderColumns(#Renamed Columns2,{"Title.1", "Rank", "Year of release", "Rating", "Title (Rank)"}),
#Reordered Columns2 = Table.ReorderColumns(#Reordered Columns1,{"Title (Rank)", "Year of release", "Rating"}),
#Removed Columns1 = Table.RemoveColumns(#Reordered Columns2,{"Title.1"}),
#Removed Columns1
```

Paste your M steps in under the already generated code. Remember to add a comma in front of the first line!

Refresh your connection and admire your newly updated list. Sadly there is no short-cut for changing the report visual - you will need to recreate this:

Title (Rank)	Year of release	Rating
The Wizard of Oz (1)	1939	0.98
Citizen Kane (2)	1941	1.00
Get Out (3)	2017	0.99
Mad Max: Fury Road (4)	2015	0.97
The Third Man (5)	1949	0.99
Moonlight (6)	2016	0.98
The Cabinet of Dr. Caligari (Das Cabinet des Dr. Caligari) (7)	1920	1.00
Inside Out (8)	2015	0.98
All About Eve (9)	1950	1.00
Metropolis (10)	1927	0.99
Total		9.88

It is also worth ordering the **Title (Rank)** column using the **Rank** column.

Optionally save this as **Out with the old, in with the new** and then close it down.