Pollution report documentation

Personal project that I took up to practice and learn how to work with Power BI better, this document folder serves as a step-by-step overview of what I did while working on the project with the data provided. This project is about government pollution in the UK, with the data being sourced directly from gov.uk, making cleaning the data essentially unnecessary in this case. This is more of a reference as to how I got my report working and functioning, along with the way I think data should be modeled and visualized for a report. For this project, I focused mainly on particulate matter, nitrogen oxides and ammonia released into the atmosphere, though there is more data present for other emissions such as methane on their website.

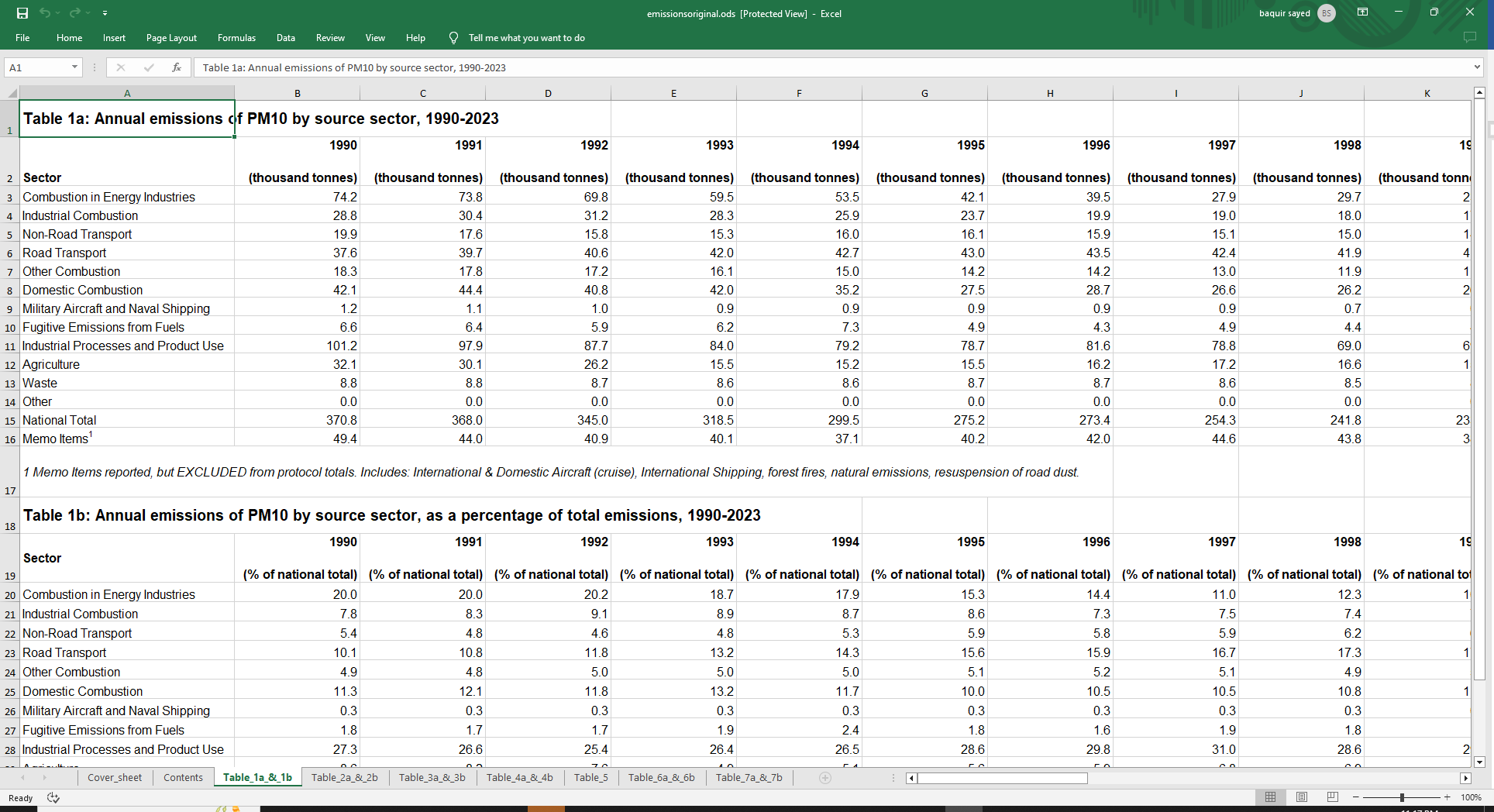
Disclaimer

I am pretty new to Power BI and data analysis in general, this was a personal project I did for practice and how to get a set of data that is extremely convenient to work with. That is probably why if someone sees the .pbix file included in the repository and checked data transformations, it ***WILL*** look unclean and would see a ton of steps that exist for almost no reason. At the time I’m writing this (3am, 06/03/25), I haven’t refactored the entire transformation process yet, but I’m still applying for jobs and such so this disclaimer is somewhat important. Using this to learn, I thought about changing the logic of my data ***3-4 times*** because I would end up finding a more efficient or better method of doing it. In this point in time (3am, again), it isn’t actually more efficient from a transformation point-of-view, but it makes it infinitely easier to visualize. Again, will change the entire transformation step at some point in the future to make pre-processing more efficient, but if you’re reading this before that has happened, then you know why the transformations are the way they are. As time goes on, this document will be updated and should ideally reflect documentation for the final product, with updated screenshots.

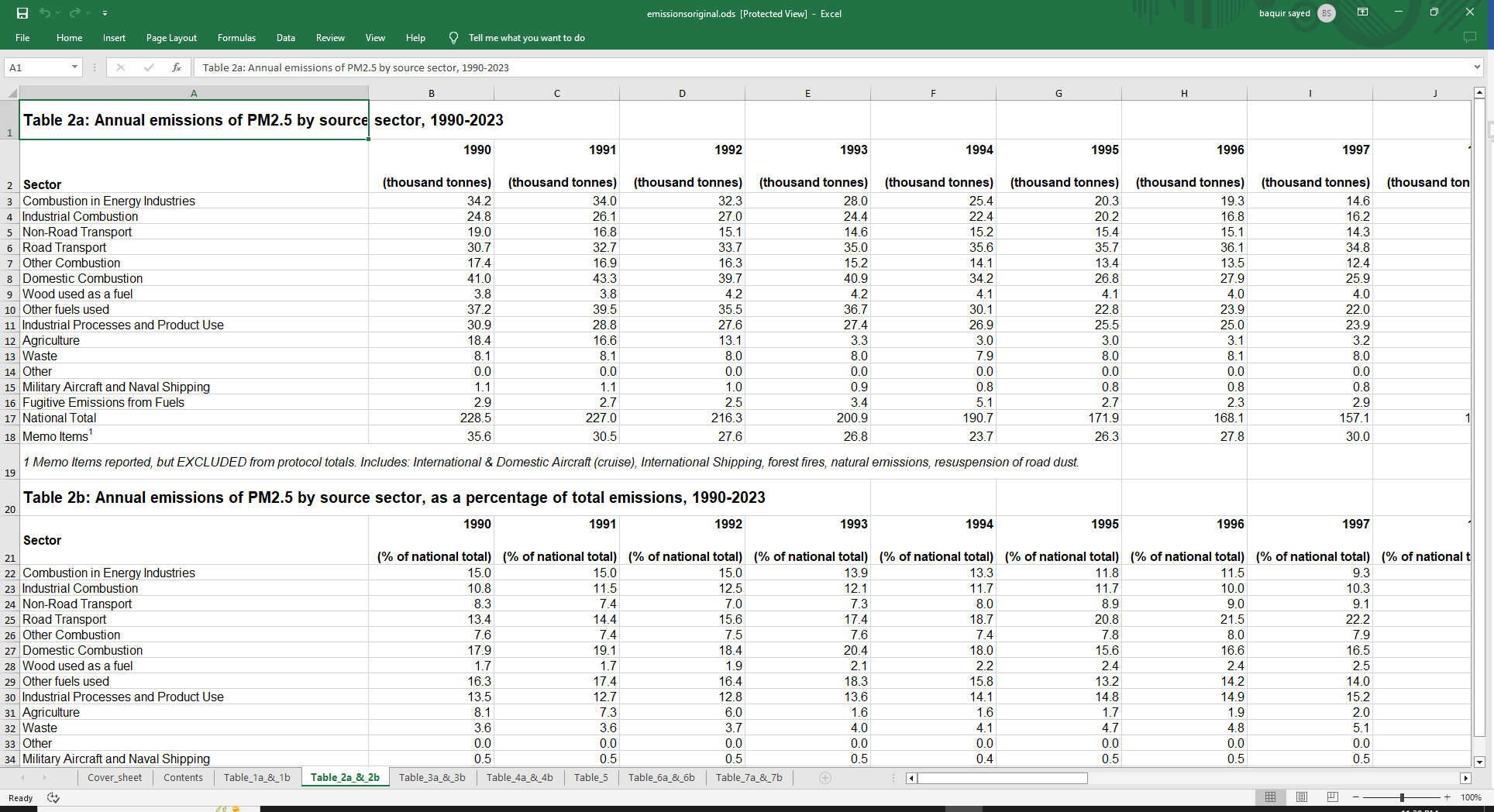
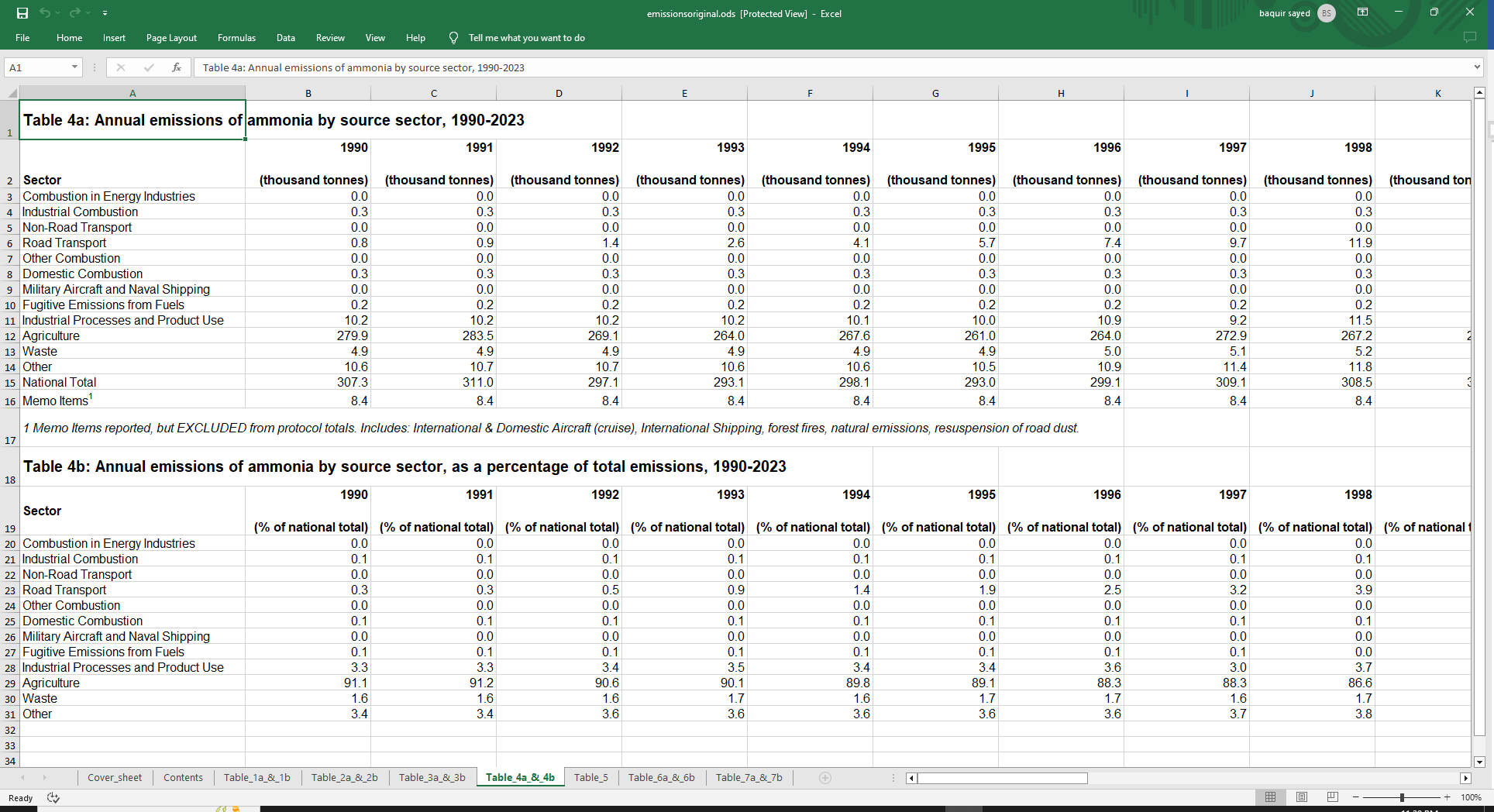
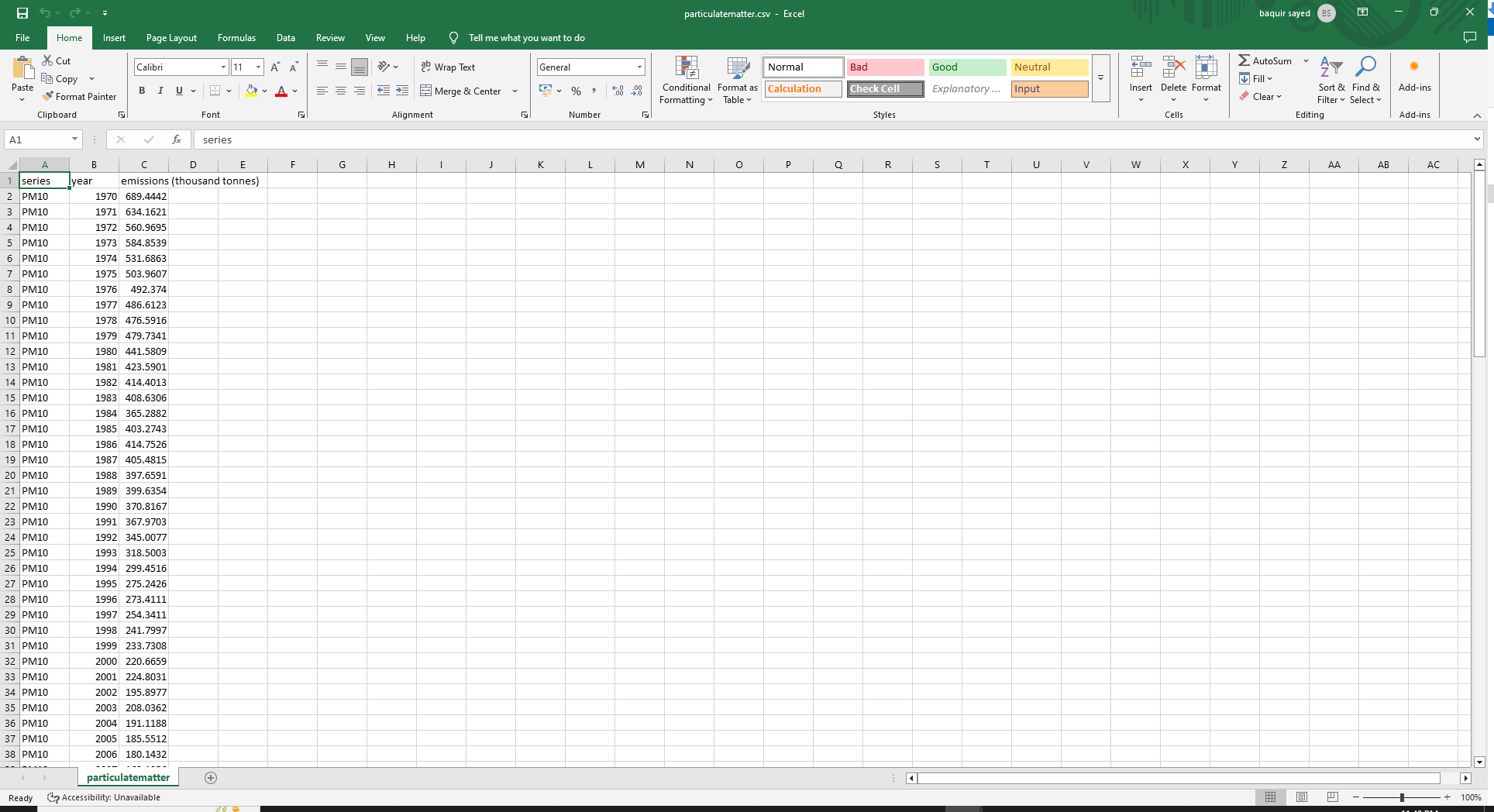
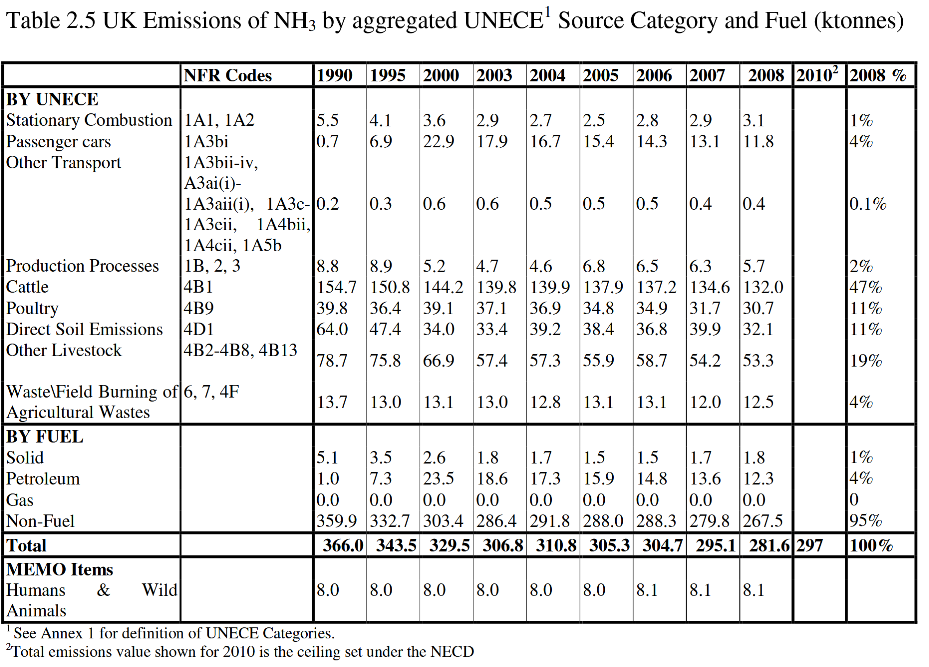
# Data sourcing, cleaning and transformation

Being directly sourced from the government, the data is very clean, so I didn’t have to do any cleaning on my end (although there is some discrepancy as mentioned later in this segment). As for transformation, I did a few transformations to make it easier to work on the data. Starting off, the data looked like this from the Excel and ODS files (links are found in reference and also uploaded to the repository). Here is an example.

**Emissions of PM10 in tonnes and as a %**

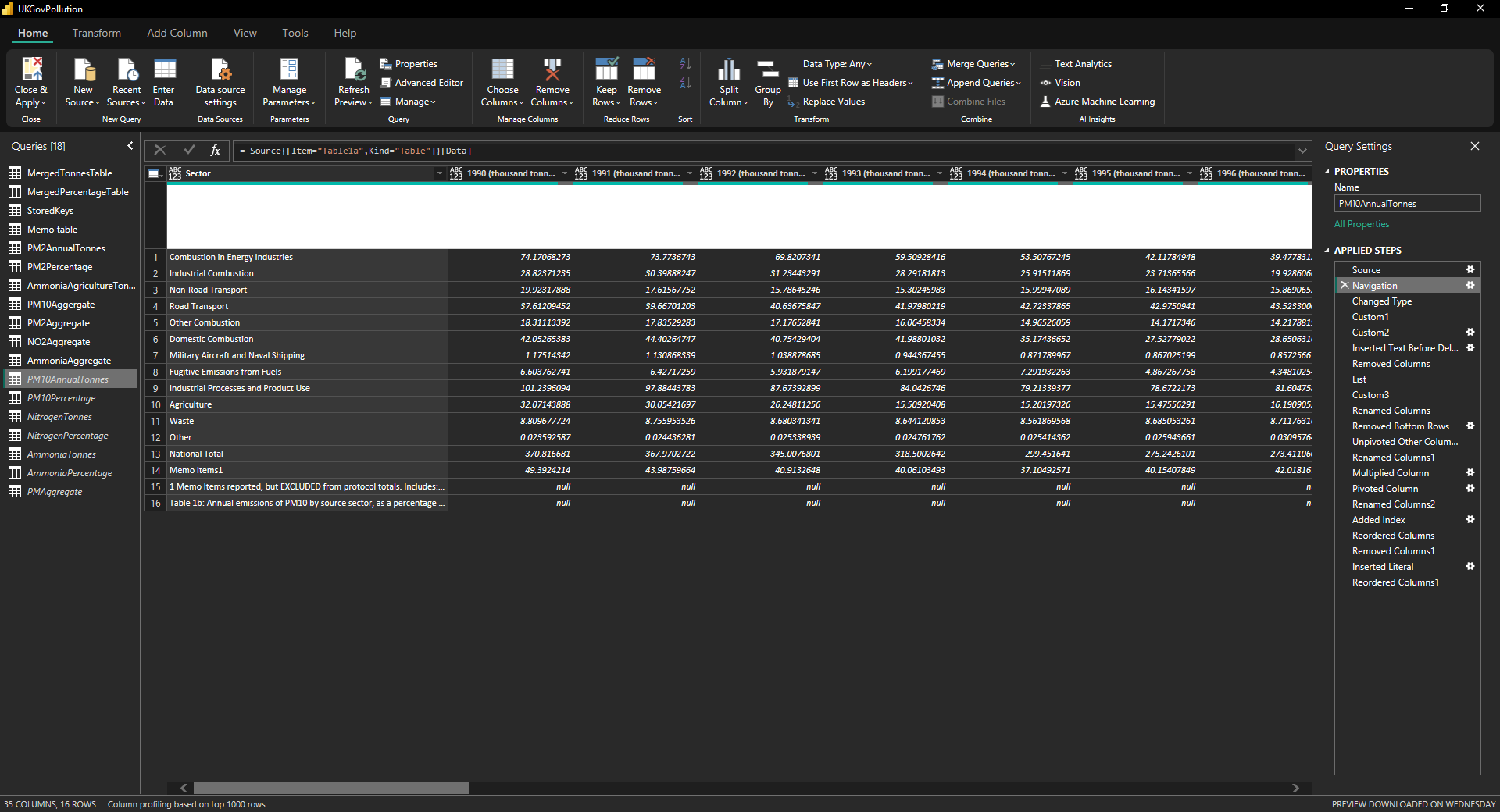


The data was split between different industries/sectors where the emissions were coming from, and were summed up yearly with one of the columns being a “National Total” for that year. There was also a list of “Memo items”, which were recorded but not included in the total, as it included data from emissions such as forest fires, international aircraft/shipping and natural emissions. From the three emissions I decided to visualize, there were a few key points that I learned while looking over the dataset.

* PM10, NO2 and Ammonia had the same sectors between the three of them, while PM2.5 and Ammonia from agriculture had different sectors. PM 2.5 had *most* of the same columns, but had a few differences, such as the inclusion of “Wood used as a fuel” and “Other fuels used”. 
* Ammonia was heavily skewed in its distribution, as most of its emissions come from agriculture. More specifically, animals and use of inorganic fertilizer produces an immense level of emissions. Between 1990 and 2023, the lowest amount of ammonia produced by agriculture was 84.3% of the national total, and thus it needed its own separate second table based only around agriculture and ammonia produced there. 
* Nitrogen Oxide was by far, the highest emission produced from the three I selected, and showed the steepest decline compared to the rest.
* The UK government had published a set of aggregate data as well, these were just the national totals in their own .csv files. The problem here is that these records stretch back all the way to 1970 (or 1980 for Ammonia). We do not have the data spread between sectors from that time period, just its aggregate. I still decided to import this data, mainly for static visuals if necessary because applying a filter context to it would be useless.   
  
* While trying to find pre-1990 sector data, I came across a report from DEFRA (Department for environment, food and rural affairs) and these reports came up with significantly higher ammonia levels and significantly lower NO2 levels from agriculture compared to the one on gov.uk. I’ve still decided to go with the data on gov.uk, mainly because it is a more complete dataset, but I’ve linked the entire report in references and a list of NFR codes aswell. The report has completely different numbers compared to everything.
* Nitrogen Oxide is measured as an aggregate in (million tonnes) instead of the standard (thousand tonnes) that is done by all other emissions, while the statistical report uses thousand tonnes.

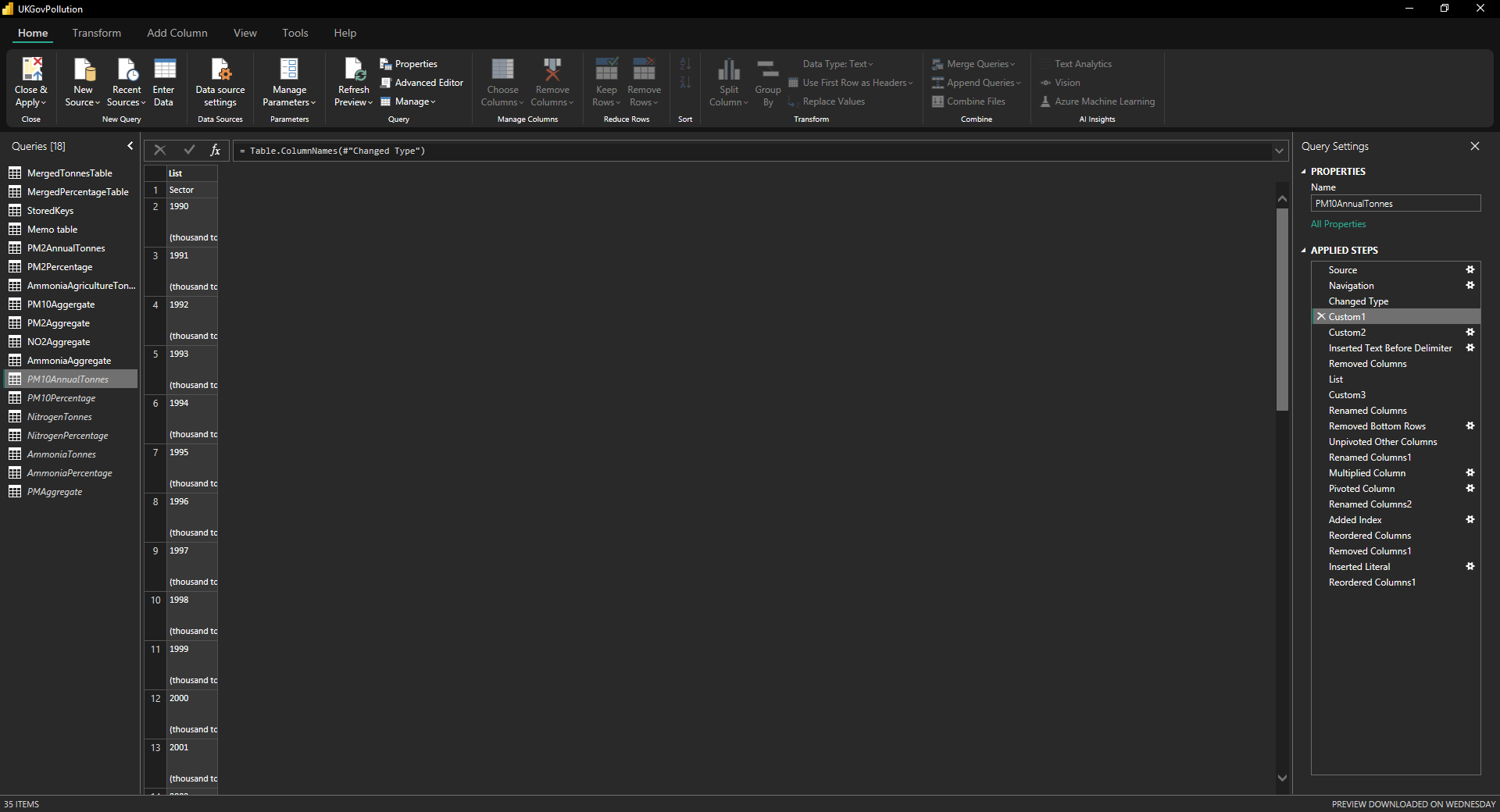
These were the first important bits of information I gleamed over while performing some basic EDA on my data and trying to find additional data sources. I tried finding data based on the sector pre-1990, but didn’t find anything thorough enough to use in a report.

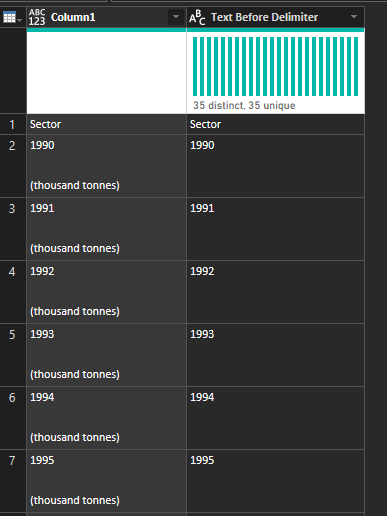
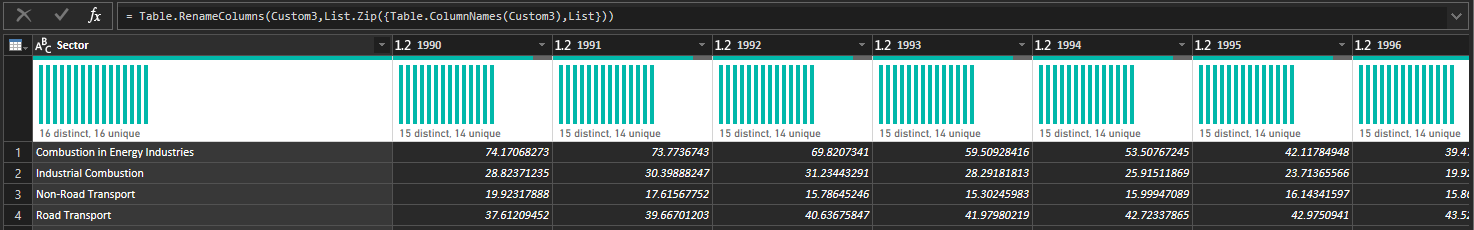
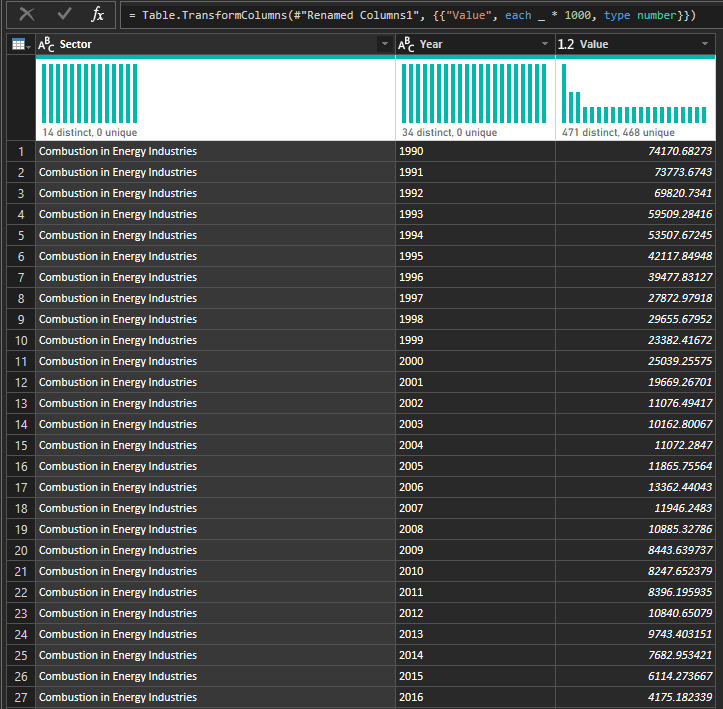
Importing data from the dataset, it conveniently split up between different tables inside the sheet itself. For example, Table 1a and 1b contain data about PM10 emissions, 1a is PM10 emissions in kilo tonnes, split based on their source sector, while 1b is percentage of the emissions split based on source sector. This allowed for me to split up the percentages and the of emissions in kilo tonnes for easier visualization and reporting. First import into power query looked like:



Theres a few problems here that are obvious from the data import, mainly:

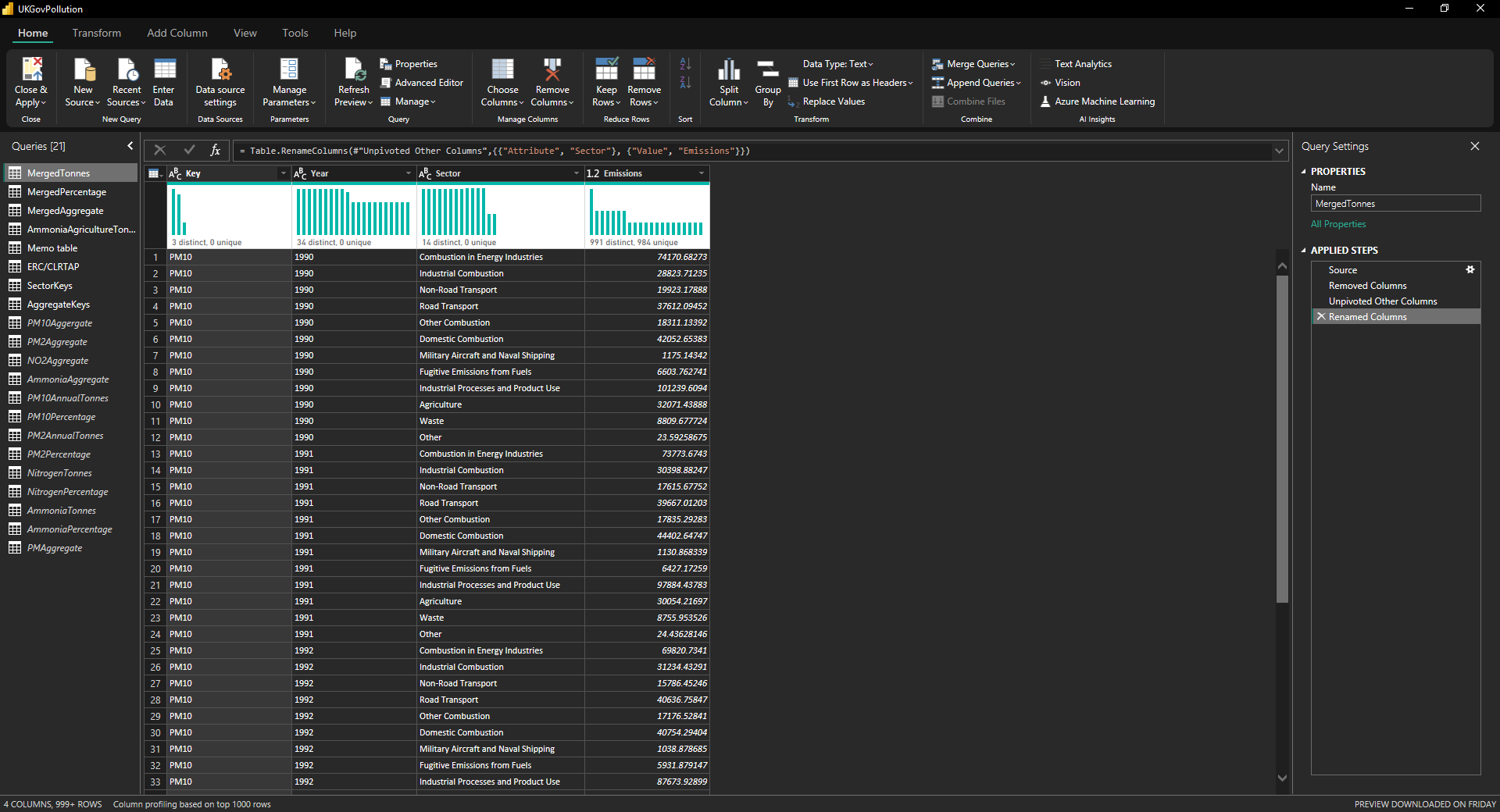
* The imported data contains two lines that aren’t data points. These are swiftly removed by removing the bottom two rows of the table.
* ~~Memo tables are also present here, while these are good to have, they present a bit of a problem because they’re not calculated in the national totals. They should be in their own table, ideally~~ After giving it some thought, there is no reason to split these up and visualize the data accordingly. More of my thought process later in this document.
* National totals are present here as well, even though we already have aggregate tables in our model. This is duplication of data and is making the size of our model larger for no real reason.
* The years aren’t written as years, but as the year (thousand tonnes). This forces them to be text instead of date, but even then I would create my own date/time table using the CALENDAR() DAX function instead.
* The values, while consistently in thousand tonnes in the statistical dataset, aren’t a good format to visualize data in. The problem here lies in the fact that because they are in thousand tonnes, and Power BI automatically compresses numbers to fit the right amount, the end-user would have to calculate twice in their head for the numbers to make sense. An example would be Ammonia, calculated in 300 thousand tonnes. The visual itself would make it look like a value between 250-350, but the user would need to know that it is in thousand tonnes. Another problem is the nitrogen aggregate calculation being in millions tonnes, compared to everything else being in thousand tonnes. The best solution here would be to get rid of both millions and thousand tonnes, and let Power BI visuals automatically compress them in the visual.

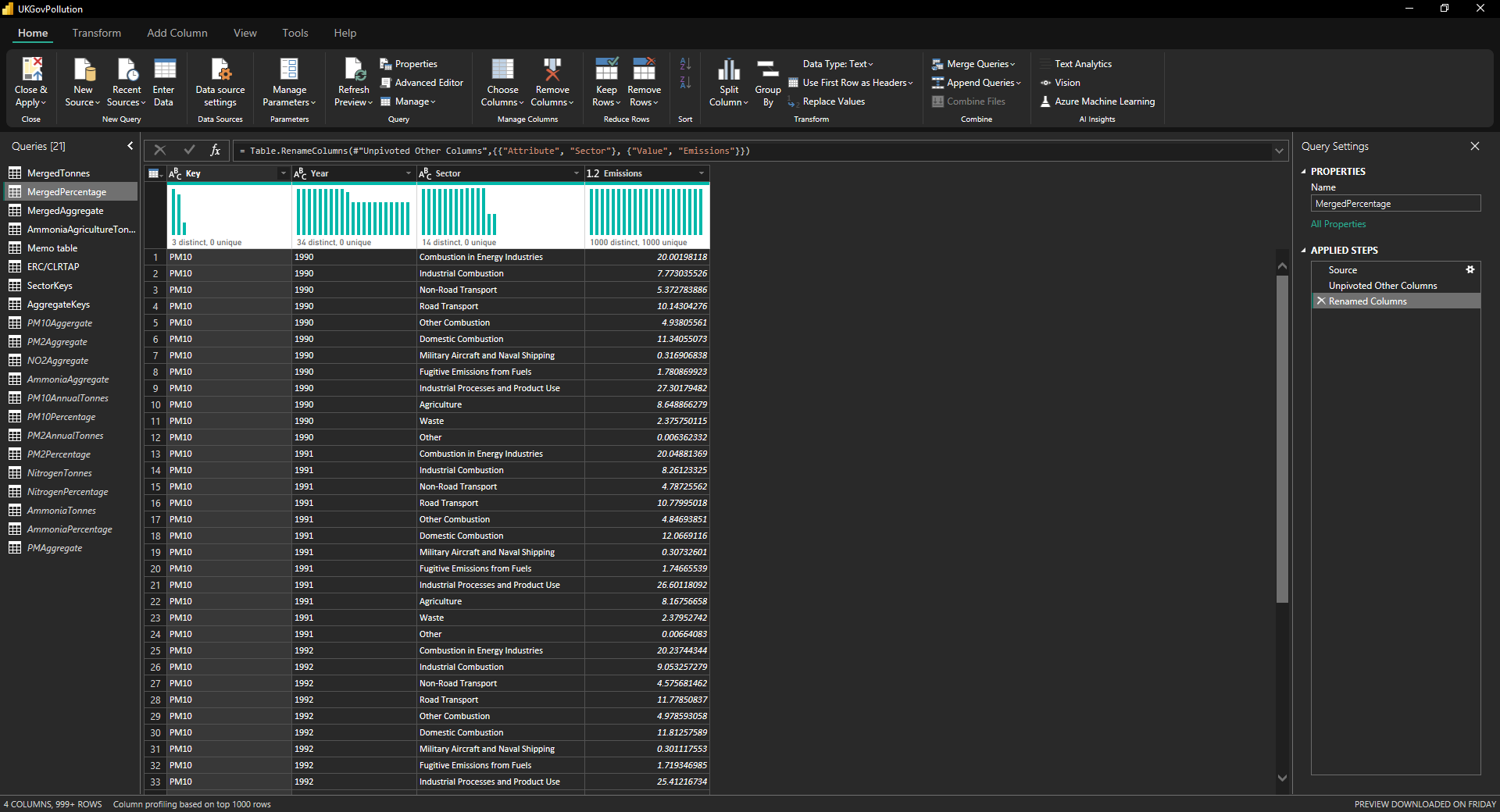
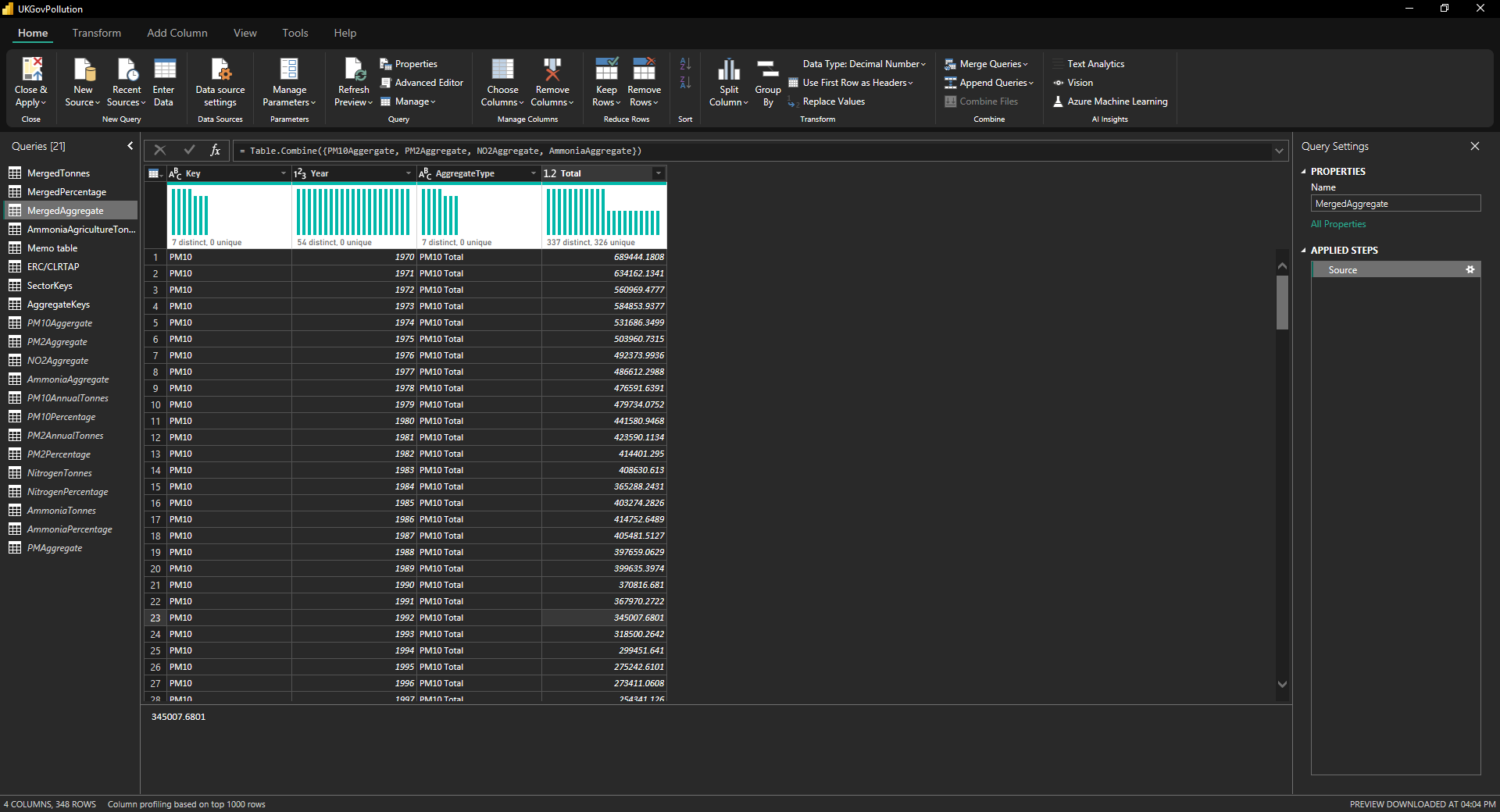
The first step I took in data transformation was to remove the thousand tonnes from all years, make them into a format that’s better for visuals. The first time around, I used a bit of M code to try to transform the data, as changing it one by one is extremely slow. I used Table.ColumnNames to produce a list of the name of every column in a table.  


I turned this into a table, then added a second column. The second column was made by using “Column From Examples” from the previous column, where I just removed the thousand tonnes and it caused it to create a new column with the updated column names.  
  
  
I then deleted the previous column, turned the updated column into a list using more M code and renamed the entire table using the Table.RenameColumns with List.Zip to rename the entire table at once.  
  
  
After doing this for a couple tables, I realized that because I wanted to merge some of the tables and make all of these into one table anyway, I would have to change the format from wide to long anyway, at which point I could just use replace values and trim to remove the thousand tonnes later during transformation, though this did teach me how to use M code to transform column names quickly.   
  
During transformation, I also multiplied all values by 1000, which was one of the problems in the data I wanted to fix   
  
After a bunch of transformations, thinking about what would work well and what wouldn’t, I finally   
came up with a set of tables that I would use.  
  


I merged all of the smaller tables into three merged tables:

* MergedTonnes: This table contains values for PM10, PM2.5, NO2 and Total Ammonia per year per sector. Merging all the tables into one is best practice as it is the closest I can come to a star schema like design, though this data isn’t very suitable for it unless you merge everything together. The merged table also includes the Memo table, because after a bit of thinking, I realized I didn’t really need the memo table to be different than the Merged Tonnes table, especially if the user can sort by Sector. Ideally, the user uses the aggregate tables for the national total, instead of using the sector total for the same job. I will adjust the visualizations accordingly to better fit that model.



* MergedPercentage: This table contains percentage values for PM10, PM2.5, NO2 and Total Ammonia per year per sector. These emissions are in decimal numbers, but are already percentage values. This means we don’t have to do any calculations for it.  
  
* MergedAggregate: This table contains totals for all emissions. These totals are of many types. For example, in aggregate type there are two entries per year for NO2. These are “NO2 Total” and “NO2 Compliance Total”. The former is the total level of nitrogen oxides released in the atmosphere, while the compliance total is as required by the National Emission Ceilings Regulations, which exclude agricultural sources for their assessment of reductions. Both of these can be important for analysis, and thus are included. The ERC limits have been taken out of these aggregate tables and made into their own table called ERC/CLRTAP and have their own set of keys for easier filtering.  
  

I also have separated agricultural ammonia, mainly because I don’t want to have all of its different sources mixed in with the sources from the merged tables. PM2.5 got away with it because it only had 3 columns like that. It will get its own filter and visualization for easy analysis. With that being done, we are ready for making a few measures and start visualizing data.

# Data Visualization

# References

<https://www.gov.uk/government/statistics/emissions-of-air-pollutants/emissions-of-air-pollutants-in-the-uk-nitrogen-oxides-nox>

<https://www.gov.uk/government/statistics/emissions-of-air-pollutants/emissions-of-air-pollutants-in-the-uk-particulate-matter-pm10-and-pm25>

<https://www.gov.uk/government/statistics/emissions-of-air-pollutants/emissions-of-air-pollutants-in-the-uk-ammonia-nh3>

<https://www.gov.uk/government/statistical-data-sets/env01-emissions-of-air-pollutants>

DEFRA report: <https://uk-air.defra.gov.uk/reports/cat07/1009030925_2008_Report_final270805.pdf>  
NFR codes: <https://naei.energysecurity.gov.uk/sites/default/files/cat07/0910130851_DA_AQ_Inventory_Report_2007_Appendices_Issue_1.pdf>