# Overview

Although all noise meters with a Type specification must adhere to set standards and criteria, there is currently no standard format in which noise measurement data is downloaded onto an electronic device (generally a PC). It is not a surprise then that each brand of meter’s download capability will serialise data to its own format. Some manufacturers will sell data processing software which can be sold to practitioners who have already bought a noise meter.

It is however the case that manufacturers will support either a direct or indirect method of transforming data into a machine-readable format known as csv (Comma Separated Values). Csv is very useful to anyone analysing noise data as it can be imported into Spreadsheet programs such as the ever-popular Excel from Microsoft. When a csv is imported into a spreadsheet each “field” from a csv file will reside in a “cell” which can be processed by a computer. Generally, for noise practitioners this processing will involve some form of aggregation (log averaging, minimal, maximum and modal values over a set period of logging). Spreadsheets are also useful for the presentation of noise data as they support graphical and pivot table functionality.

Csv files are useful for serializing data from a period of continuous measurement such as is generally required when a practitioner undertakes environmental noise monitoring as such data will typically contain a large number of discrete measurements logged at set intervals over a period of time. This type of data is known to data analysts as **Time Series** data.

A csv file of time series data will contain a row representing each period of time that a noise measurement has been logged. Each row of data will as a minimum contain the start or end date and time of measurement and a series of noise metrics. Typically for UK environmental noise the metrics most of interest are the “energy” average noise level (LAeq), the background noise levels (LA90) and the maximum impulsive noise level (LAmax,F). Using these metrics practitioners aggregate levels over periods of time in order to characterise the noise levels at a noise monitoring location. A sample **Time Series** format for noise data is shown below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date** | **Time** | **LAeq** | **LAmax** | **LA90** |
| 29/11/2024 | 13:15:00 | 54.9 | 78.3 | 51.3 |
| 29/11/2024 | 13:30:00 | 54.4 | 60.7 | 51.5 |
| 29/11/2024 | 13:45:00 | 54.3 | 61.4 | 50.3 |
| 29/11/2024 | 14:00:00 | 55.1 | 63.7 | 51.6 |
| 29/11/2024 | 14:15:00 | 54.5 | 74.3 | 51.2 |
| 29/11/2024 | 14:30:00 | 54.3 | 60.7 | 51.2 |
| 29/11/2024 | 14:45:00 | 58 | 77.2 | 52.1 |

Whilst it is reasonable to assume the above it is unfortunately the case that, that is all that can one assume. Every meter from each manufacturer and sometimes even different versions of download software for the same meter may differ in the csv format that may come to the user.

* Dates and times may be in different columns or “merged” into “one”.
* The time may the start or end time of the logging period.
* There may be more than one header row.
* The names of the columns differ e.g. LAeq Overall, Leq A and the orders in which they appear from left to right.
* There may be non-data rows placed amongst valid data, particularly if it is presented in and old style “print” format.
* There may be summary rows at the foot of the csv tables.

So, a practitioner who may use several different noise meters in the course of their work is forced to undertake a custom step in which the output from the meters is “normalised” into a common format. With this step accomplished the data can be processed as required for the job in hand.

It would therefore be useful for practitioners to have a software tool that they could use to “normalise” the csv format from any meter into a common format, and thereby save the trouble of doing so manually. If possible, it would be useful if it could perform the aggregations required by environmental noise practitioners who could customise the aggregation steps to their own use.

It would be perfectly possible to write a program to carry out the above. Indeed the author of this article has done so for his own consultancy. However assuming a Microsoft-only solution is acceptable , Excel and PowerBI offer a feature called **Power Query** which is generally fit for purpose for carrying out the tasks noise practitioners require.

Power Query enables users to extract, transform, and load (ETL) data from various data sources, including the one relevant for us i.e. csv. The following three features are pertinent to its use in PQforNoise.

* Data Transformation - Clean, filter, and reshape the data using a variety of transformation tools like pivot, unpivot and merge;
* Data Loading - Load the transformed data into Excel or Power BI for analysis, visualization, and reporting;
* Automation - Save the steps of your data transformation process as a query, which can be refreshed to update your data with the latest information.

While it is for the most part a “no-code” tool it does accept custom functions and these were necessary to successfully aggregate data in this case. There is a learning curve with PowerQuery and a greater one with Power BI but it is much less than would be required for software development. So, it is expected that practitioners will heavily customise it for their own purposes.

# Preparation

## System Requirements

You will needs a PC running Windows 10 or later. Excel 2016 or later should be installed (or as part of Microsoft 365).

The PowerQuery transformations are generally CPU bound so a powerful CPU will mean that they take less time.

## Contents of repository

The repository contains the following

* This Readme document
* PQforNoise.xls – This spreadsheet contains the PowerQuery set-up and initial settings of parameters.
* SVAN971.csv, CirrusCR821A.xls ,Larson820.xls and Nor140.csv – 4 sample csv files from different suppliers.

Copy all files to a folder on your PC.

## Starting

To operate it do the following.

Open the csv file of interest and also the spreadsheet **PQforNoise.xls.**  Go to the Parameters worksheet.

**Note** – it may be necessary when changing values to hit **F9** to manually update the cells as automatic update is turned off in PQforNoise for performance reasons.

## Configuration

### Period Table

The Periods Table defines the day and night periods used in environmental assessments. It’s initial setting is daytime from 07:00 to 22:59, night-time inclusive from 23:00 to 06:59 with the data logging times rounded down to the nearest minute. This is the default for most environmental noise in the UK.

**Note**, you may need to redefine these if you have an “Evening” period or if your csv details end times rather than start times of logging (this is quite rare but does occur in some formats).

### Run Parameters table

This contains the following:

* **Average Aggregation Interval**

This allows you to aggregate LA90 and LAeq values over a longer period than the default. e.g. If each row of your csv data is at 1 minute intervals and you want to aggregate LAeq and LA90 over a 15 minute period, you would enter 15 In effect the transition will aggregate by logarithmically averaging over the interval period.

**Note:** Logarithmically averaging LA90 values is not mathematically correct but may approximate to a real LA90.

* **Maximum Aggregation Interval**

As above. You can decide the interval for aggregation of LAmax values. In practice this will take the highest LAmax value recorded in the aggregation period.

* **LAMax Rank**

Environmental noise assessments generally look at the 10th highest LAmax value recorded over a night-time period so the default here is 10. A different value can be selected if required.

* **Path and Name**

Path and name to the csv file. The path expects a backslash “\” on the end e.g **“c:\data\”.**  I have not included a file browse for file button here as I did not want to put any VBA into the spreadsheet. However it is straightforward to do of you have knowledge of VBA.

### Labelling the csv files

The Power Query transformation searches the csv file for labels that define the location in the time series your columns are and where they start. The labels are:

* Either “**\\*DateandTime\*\”**  or “**\\*Date\*\”** and “**\\*Time\*\”** depending on the csv file format.
* **\\*LAeq\*\**
* **\\*LAmax\*\**
* **\\*LA90\*\**

An example is shown below where labels define where the time series date starts. I have highlighted them in red for clarity here (they don’t need to be red).

A screenshot of a computer

AI-generated content may be incorrect.

So, for each csv file you wish to process you must add those labels to the csv file above the data. Four sample files are in the enclosed zip file.

# Operation

## Process

With the above complete click **“Data”** then **“Queries and Connections”** from the toolbar. You should see the screen below.



The 3 Data Tables at the top right are created by the transformation. You can see them on the worksheets **“Summary”,** **“Background Analysis”** and **“Time Series”.** The other functions and queries shown are used by Power Query internally to produce these 3 tables.

On the top menu click **“Refresh All”.** Depending on the size of your csv file it may take a little while to process and you can track progress in the right-hand pane. Once complete Power Query should process the results and update the 3 tables on the top right.

## Results

### Time Series

This is a time series view of the data processed. This table shows the raw data used by the time series graph, aggregation table and pivot chart.

### Background Analysis

A simple pivot chart showing the modal background noise level over an assessment period.

### Aggregated Summary Table

A table showing the usual aggregations for an English environmental noise assessment.

### Time Series Graph

A simple time series line chart showing the 3 main metrics over the monitoring period.

### Errors

The number of errors are also reported in the right-hand pane. To see an example of this Select **CirrusCR821A** format and Csv filename **CirrusCR821A**. You should see 7 errors reported in **AggregatedSummaryTable**. You should see 7 errors reported. The errors occur because there were less than 10 **Lamax10** events in 7 of the periods. It is hoped that as you increase your familiarity with Power Query you should find it relatively straightforward to diagnose such errors.

# Under the hood

To understand what Power Query is doing under the hood go to the Power Query Editor **“Data”** then **“Get Data”** then **“Power Query Editor”.**

Click on **“Time Series Table”** then **“View”** from the top menu then make sure **“Query Settings”** on the left of the toolbar. You can see a list of each Power Query Step that has been applied to transform the data into the tables you can see. All of the transformations are standard ones which have been added using the UI. You can work these out by looking at the Power Query documentation. Some of have been modified slightly to look at the parameter tables we have already discussed and there is a small amount of custom code.

# The Future

I am hoping that practitioners will be able to use this as is but expect that in reality they will customise it for their own purposes. It would be nice if there was some collaboration going forward for functionality which benefits all consultants and hence I have placed it into github to give it an eternal life. There are many mods for the future. The most useful would be to change the functionality that relies on positional parameters for column locations to parameters that can detect it by header contents. This is not easy to do given that there may be multiple header rows and for example more than one marked “LAeq” (“LAeq Overall” and LAeq 125 Hz”). A whitelist and blacklist approach to contents is probably the way forward. But it is probably doable and likely the next change when time allows.

# Terms of use

The PQforNoise was built by Sean Sullivan of Sulltech Ltd.

Neither I nor Sulltech Ltd accept any liability for its results. You use it at your own risk, and you should carry out your own checks and balances to ensure you are satistfied that the output is accurate.

This is an open source project, and I welcome suggestions for changes, improvements or new features. You can also write your own methods or functions and share them with me, either by getting in touch offline, or by creating a new branch from this Git repository.

You may use this toolkit subject to the licence conditions below. For clarity, you may use this toolkit or adaptations of it in your day-to-day engineering work, but incorporating it into a commercial software product or service is not permitted.

This project is being shared under a [Creative Commons CC BY-NC-SA 4.0 Licence](https://creativecommons.org/licenses/by-nc-sa/4.0/).

Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

NonCommercial — You may not use the material for commercial purposes.

ShareAlike — If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.