Eionet Report - ETC/ATNI 2021/6

Noise indicators under the Environmental Noise Directive 2021. Methodology for estimating missing data

July 2021

Authors:
Jaume Fons-Esteve (UAB), Francisco Domingues (UAB), Maria José Ramos (UAB)

ETC/ATNI consortium partners:
NILU – Norwegian Institute for Air Research, Aether Limited, Czech Hydrometeorological Institute (CHMI), EMISIA SA, Institut National de l'Environnement Industriel et des risques (INERIS), Universitat Autònoma de Barcelona (UAB), Umweltbundesamt GmbH (UBA-V), 4sfera Innova, Transport & Mobility Leuven NV (TML)

European Environment Agency
European Topic Centre on Air pollution, transport, noise and industrial pollution

Cover design: EEA
Layout: ETC/ATNI

Legal notice

The contents of this publication do not necessarily reflect the official opinions of the European Commission or other institutions of the European Union. Neither the European Environment Agency, the European Topic Centre on Air pollution, transport, noise and industrial pollution nor any person or company acting on behalf of the Agency or the Topic Centre is responsible for the use that may be made of the information contained in this report.

Copyright notice

Information about the European Union is available on the Internet. It can be accessed through the Europa server (www.europa.eu).

The withdrawal of the United Kingdom from the European Union did not affect the production of the report.
Data reported by the United Kingdom are included in all analyses and assessments contained herein, unless otherwise indicated.

Author(s)
Jaume Fons-Esteve, Francisco Domingues, Maria José Ramos (UAB)

ETC/ATNI c/o NILU
ISBN 978-82-93752-29-5

European Topic Centre on Air pollution,
transport, noise and industrial pollution
c/o NILU – Norwegian Institute for Air Research
P.O. Box 100, NO-2027 Kjeller, Norway
Tel.: +47 63 89 80 00
Email: etc.atni@nilu.no
Web : https://www.eionet.europa.eu/etcs/etc-atni

Contents

Summary

The Environmental Noise Directive (END) offers a common approach to avoiding and preventing exposure to environmental noise by reporting noise mapping and action planning. This leads to reducing the harmful effects of noise and preserving quiet areas. Noise sources, as defined by the END, include major roads with more than three million vehicle passages a year; major railways with more than 30 000 train passages per year; major airports with more than 50 000 movements per year (a movement being a take-off or a landing), excluding those purely for training purposes on light aircraft; and noise from roads, railways, airports and industries inside of agglomerations - part of a territory, delimited by the Member State, having a population in excess of 100 000 persons and a population density such that the Member State considers it to be an urbanised area. It is important to note that the directive does not set limit values but reporting thresholds.

The END requires Member States to determine the number of people exposed to noise sources, inside and outside urban areas, and large industrial installations inside urban areas, using 5 dB interval bands at L_{den} ≥ 55 dB and at L_{night} ≥ 50 dB. The experience of the three reporting cycles of the END, which started in 2007, has demonstrated substantial delays in delivering noise exposure data. For that reason, methodologies to estimate missing data have been developed as early as 2013 to have a complete overview of the extent of the population exposed to the noise sources mentioned above sources. This report reviews previous methodologies and provides a comprehensive description of the method to gap-fill missing data in all reported noise sources. In particular, it provides a systematic approach when regression is required (e.g. estimation of the population exposed to agglomerations roads based on the number of inhabitants) and explore a new method for major airports based on estimating the noise contour band when this information is not provided.

Acknowledgements

Eulàlia Peris, the EEA project manager, supported the work and provided valuable comments on the report.

1 Introduction

The Environmental Noise Directive (END) (EU, 2002) offers a common approach to avoiding and preventing exposure to environmental noise by reporting noise mapping and action planning, thereby reducing its harmful effects and preserving quiet areas. Noise sources, as defined by the END, include major roads with more than three million vehicle passages a year; major railways with more than 30 000 train passages per year; major airports with more than 50 000 movements per year (a movement being a take-off or a landing), excluding those purely for training purposes on light aircraft; and noise from roads, railways, airports and industries inside of agglomerations -part of a territory, delimited by the Member State, having a population in excess of 100 000 persons and a population density such that the Member State considers it to be an urbanised area. It is important to note that the directive does not set limit values but reporting thresholds.

The END requires the Member States to determine the number of people exposed to the noise sources, inside and outside urban areas, and large industrial installations inside urban areas using 5 dB interval bands at L_{den} ≥ 55 dB and at L_{night} ≥ 50 dB. The experience of the three reporting cycles of the END (2007, 2012, 2017) has demonstrated substantial delays in the reporting of the Member States, a consequence of a learning process dealing with the complexity of the reporting and the END requirements. For that reason, methodologies to estimate missing data have been developed as early as 2013 (Jones, 2013) to have a complete overview of the extent of the population exposed to the noise sources.

The latest update of the methodology was reported by Ramos (2019), where the need for some improvements was identified -and out of the scope of the work at that time. The current report tackles the following issues to improve the methodology and systematise the estimation of the missing data:

- Provide a systematic approach to test alternative regression models when pertinent (e.g. population exposed to roads inside agglomerations).
- Specify the methodology for the propagation of errors, leading to figures of total people exposed with the corresponding confidence interval.
- Improvements on the estimation of people exposed to noise from major airports based on the estimation of the noise contour for L_{den} 55 dB.

This methodological report summarises the steps followed to obtain estimated results of a complete noise exposure covering the END sources.

2 Input data

This report is based on the review of the methodology developed in 2019 (Ramos, 2019). Therefore, the same data source has been used, for comparability reasons, when data was required to test some improvements. The data covers the data reported until 01/01/2019([1]).

In order to facilitate the implementation of the workflow for estimating missing data, the minimum data requirements are specified in Table 2.1, Table 2.2, and Table 2.3 for the different noise sources. Therefore, a preliminary step is to extract the needed information from the internal noise database(s), which compile the reported data. The most critical part is the identification of completeness for major roads and major rails. The objective of this report is out of the scope of the preliminary data selection -details are available at Ramos (2019). Moreover, changes in the data model implemented in Reportnet 3.0 may facilitate selecting the needed data. For that reason, Table 2.1, Table 2.2, and Table 2.3 are provided as a reference for futures use.

It should be noted that the methodology also requires data from the previous reporting cycle. The notation used in this report to generalise the method, is detailed as follows:

- t_n, data from the current reporting cycle
- t_{n-1}, data from the previous reporting cycle.

Table 2.1: Structure of the input data for people exposed to noise from roads inside agglomerations

Field	Type of data	Comment
Country	String	
Country2	String	
Agglomeration_name	String	
RLID	String	
UniqueAgg_ID	String	
Year	Integer	Reference year of the reported data
Population	Integer	
L_{den} per noise band	Integer	-1 not applicable, -2 not available (reported), -9999 not reported
L_{night} per noise band	Integer	-1 not applicable, -2 not available (reported), -9999 not reported

Table 2.2: Structure of the input data for people exposed to noise from major airports

Field	Type of data	Comment
Country	String	
Mair_name	String	
ICAO_code	String	
Year	Integer	Reference year of the reported data
L_{den} per noise band	Integer	-1 not applicable, -2 not available (reported), -9999 not reported
L_{night} per noise band	Integer	-1 not applicable, -2 not available (reported), -9999 not reported

[1] https://forum.eionet.europa.eu/etc-atni-consortium/library/subvention-2019/task-deliveries-action-plan-2019/task-1.1.5.1-noise-data-operational-compilation-and-management/subtask-1.1-update-database-cws/datasets

Table 2.3: Structure of the input data for people exposed to noise from major roads and major rails

Field	**Type of data**	**Comment**
Country	String	
Country2	String	
Unique identifier(s)	*String (Several fields)*	All needed fields that provide the unique link to the noise database
Completeness	String	Complete, partial, not applicable, not provided
Year	Integer	Reference year of the reported data
L_{den} per noise band	Integer	
L_{night} per noise band	Integer	

3 END agglomerations data: gap filling

3.1 Gap filling method for agglomerations- road

3.1.1 Overview

This method is based on the working paper “Establishing a methodological proposal to interpolate a complete coverage on noise exposure at the EU level” (ETC/ACM, 2015), with some improvements described by Ramos (2019).

Figure 3.1 provides an overview of the process for estimating missing data adapted from Ramos (2019). In each step, the decision tree uses the best approach according to the available ancillary data:

- Use data from the previous reporting period if available.
- When this information is also missing, the following steps are taken:
    - People exposed is estimated with regression analysis, being the number of inhabitants the independent variable.
    - Finally, estimate the distribution of the population exposed by noise bands applying the European average.

A more detailed explanation and the basis for the methodology applied in each step is provided in Table 3.1.

The gap-filling methodology also includes those few cases where data is missing for some noise bands. Then, **partial gap filling** is conducted as follows:

- Use data from the previous reporting period for the missing noise band(s).
- If data from the previous reporting period is not available use the European average of % of the population exposed by noise band to estimate the missing data. Table 3.2 provides an example of a partial gap-filling when data from the previous period is not available.

The following sections focus on those aspects that were not documented on previous reports:

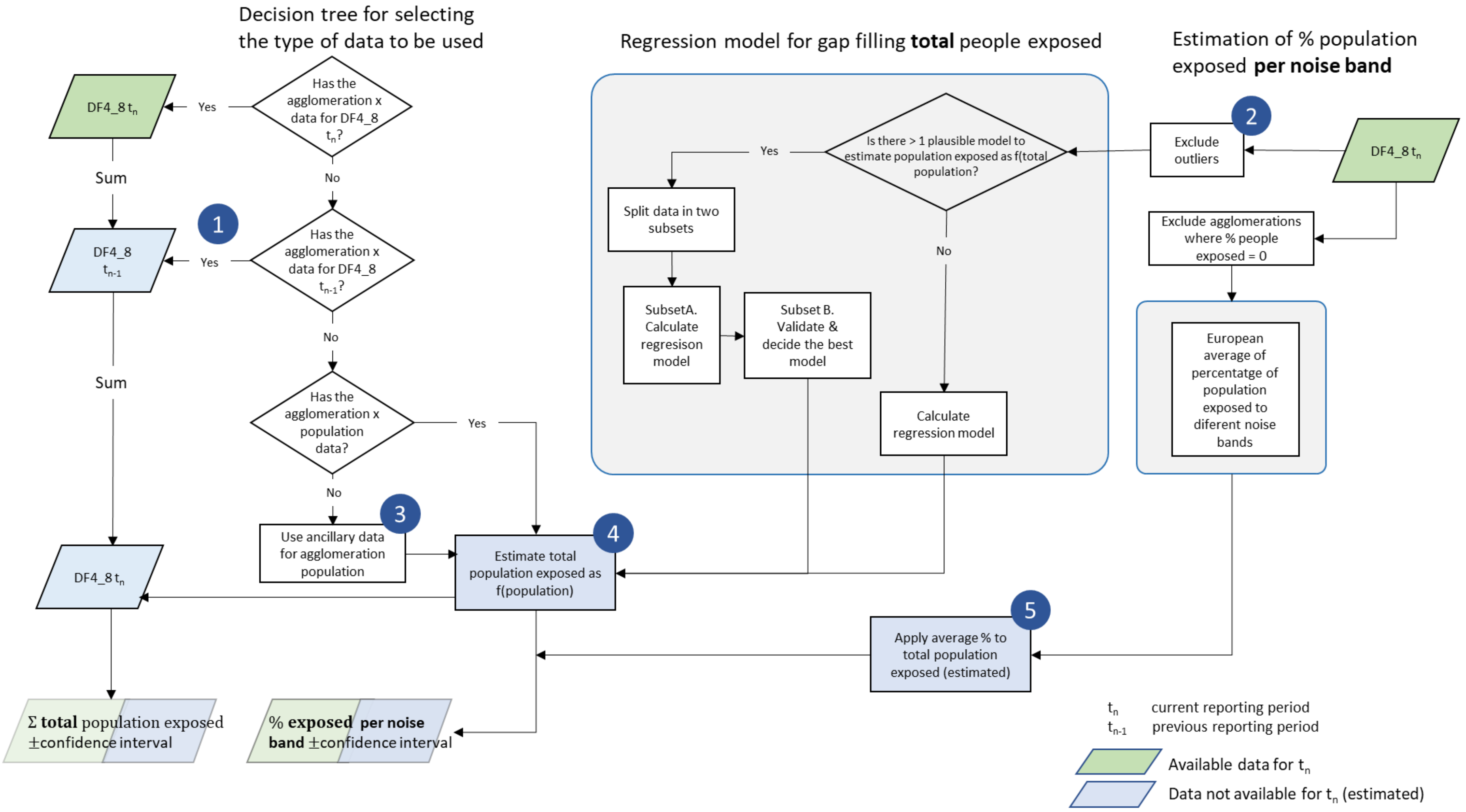
- Selection of the appropriate regression model
- Estimation of the missing data with the corresponding error (error of the prediction).
- Calculation of the people exposed in Europe as sum of all the agglomerations (reported and gap filled). Calculation of the propagation of error as a result of adding individual values with their own error (error of the prediction).
- Estimate the distribution of total noise exposure by noise bands and associated error -error of the function’s sum and error to distribute the total by noise bands.

All these steps have been implemented in R to ensure the traceability and replicability of the whole process, including the assessment of different regression models.

Table 3.1: Methods used on the various steps of gap filling people exposed to noise from roads inside agglomerations. Numbers refer to the steps highlighted in Figure 3.1

Step	Method for gap filling	Explanation	Comment
1	Use the data delivered in the previous reporting period for the same agglomeration, if available.	A comparative analysis described in Fons et al. (2017) concluded that this is the method with the smaller error.	The method is constrained to the availability of the data on the previous reporting cycle.
2	Exclude outliers from the reported data to be used for the regression (step 4)	Outliers from the percentage of change between t_{n-1} and t_n based on the interquartile range (IQR, the difference between 3rd and 1st quartile). The exact threshold is ± 1,5 IQR.	Previous assessments have identified some extreme population changes exposed to noise between two reporting periods (Ramos, 2019). Therefore, the estimation of missing data in 2019 (Ramos, 2019), adopted a methodology to exclude outliers.
3	Estimate the total population from ancillary data.	If the country reported the agglomeration's delineation, the population could be derived from Urban Atlas described in Fons et al. (2015). The average error of the estimation based on Urban Atlas is 3%, ranging from 1 to 10%. If the delineation has not been reported, data from Eurostat can be used. The agglomeration population is the independent variable of the regression (step 3) to estimate the population exposed when data from the previous reporting cycle is not available.	The reporting cycle of the Urban Atlas is always one year later than the END.
4	Estimate the population exposed from the regression between population exposed and population of the agglomeration.	The regression and correlation analysis between population exposed and potential predictors (total population, area,…) is documented in Fons et al. (2015). Later on, Fons et al. 2017) demonstrated that, if available, using data from the previous reporting period (step 1) is more accurate than the regression approach.	The current report provides a detailed description of the metrics to evaluate the best regression model, including estimating the error and confidence interval in the final aggregation of data (EU figures). There is no a priory regression model to be applied each time that the gap filling is developed. The regression model needs to be checked each time since the relationship is strongly dependent on the data included for the regression.
5	Estimate the % of the population exposed per noise bands (%) from the European average	Based on the total number of people exposed per each reported agglomeration, we calculate the percentage that each noise band represent versus the total number of people exposed, for L_{den} and for L_{night}. Then we derive the mean at European level. This approach is discussed in depth in Fons et al. (2015).	Initially, the % was calculated on a country basis when were enough agglomerations reported. The assessment was done in Fons et al. (2017) highlights that using the country average has a similar error than the European average. Therefore, it was decided to use in all cases the European average for simplicity.

Figure 3.1: Overview of the process for estimating the population exposed to roads inside agglomerations when data is not available. The methodology only applies to agglomerations that have to report according to END requirements. Numbers indicate specific methods for gap filling depending on available ancillary data -details are described in Table 3.1. Source: updated from Ramos (2019)



Decision tree for selecting the type of data to be used
Regression model for gap filling **total** people exposed
Estimation of % population exposed **per noise band**
DF4_8 t_n
Yes
Has the agglomeration x data for DF4_8 t_n?
Sum
No
1
DF4_8 t_{n-1}
Yes
Has the agglomeration x data for DF4_8 t_{n-1}?
No
Sum
Has the agglomeration x population data?
Yes
No
3
Use ancillary data for agglomeration population
4
Estimate total population exposed as f(population)
DF4_8 t_n
Σ **total** population exposed ±confidence interval
% **exposed per noise band** ±confidence interval
Yes
Is there > 1 plausible model to estimate population exposed as f(total population?
Split data in two subsets
SubsetA. Calculate regresison model
Subset B. Validate & decide the best model
No
Calculate regression model
2
Exclude outliers
DF4_8 t_n
Exclude agglomerations where % people exposed = 0
European average of percentatge of population exposed to diferent noise bands
5
Apply average % to total population exposed (estimated)
t_n current reporting period
t_{n-1} previous reporting period
Available data for t_n
Data not available for t_n (estimated)

Table 3.2: Example of partial gap-filling when data is missing for some noise bands. N.d., no data reported (missing data)

L_{den} dB bands	**55-59**	**60-64**	**65-69**	**70-74**	**>75**
Reported data	11.7680	6.000	1.500	*n.d.*	*n.d.*
% of people exposed distributed by noise band (European average)	45,8	28,3	18,3	7,0	0,6
Reported + gap filled data (italics)	11.7680	6.000	1.500	*9.500*	*800*

3.1.2 Regression

The regression procedure could be summarised as follows:

1. Identify outliers from the percentage of change of the people exposed between current reporting period (t_n) and the previous reporting cycle (t_{n-1}).
2. Plot L_{den} against the number of inhabitants to visually inspect the most suitable regression model. Since it is not always obvious which is the best model, the most plausible ones are retained and tested.
3. Transform the data if it improves the linearity (the most common transformation in previous assessments was log transformation).
4. Divide the data in two groups to test different regression models: one group is used to run the regressions, and the second group is used to validate the models.
5. Calculate the regression model and analyse the corresponding statistics.
6. Estimate the number of people exposed on the validation subset and compare results between different models.
7. Apply the selected model to the missing data
8. Calculate the total people exposed in Europe with the confidence interval corresponding to the estimated data.

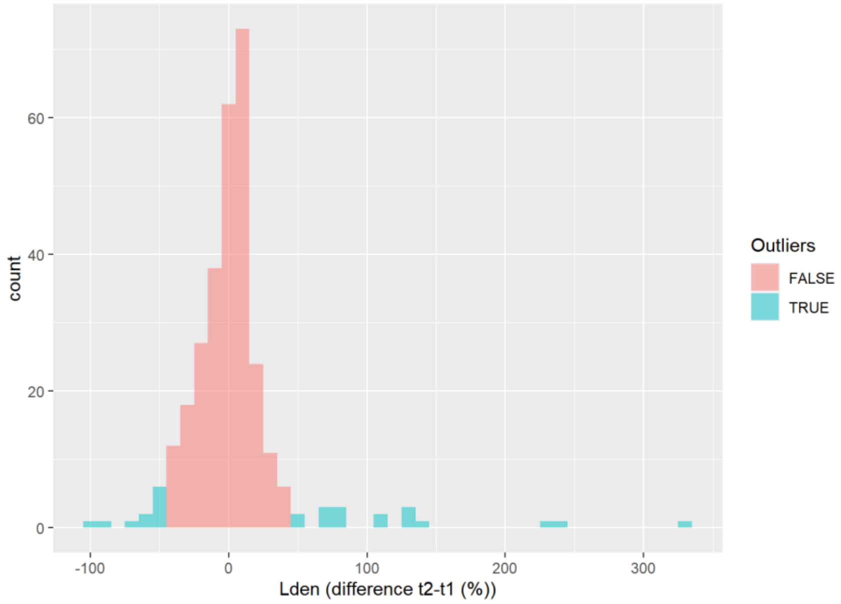
Each step is further described in the following sections. We have used the estimation of population exposed to L_{den} to illustrate the methodology. The same approach could be followed in other sources and L_{night} when required.

3.1.3 Identify outliers and adjust agglomerations with > 100% people exposed

Previous assessments have identified some extreme cases of change of population exposed to noise between two reporting periods (Ramos, 2019). Therefore, the estimation of missing data in 2019 (Ramos, 2019), adopted a methodology to exclude outliers. This methodology is based on the interquartile range (difference between 3rd and 1st quartile).

Figure 3.2 illustrates the distribution of both outliers and non-outliers for the percentage of change of population exposed to L_{den} from roads inside agglomerations equal or greater than 55 dB. In that case about 10% of the data were identified as outliers.

Figure 3.2: Histogram of the percentage of change of people exposed between t1 and t2 (roads inside agglomerations). The colour differentiates outliers from non-outliers. N = 299 agglomerations



A small number of agglomerations declared more than 100% of the total population exposed. This may be possible due to the rounding to the nearest hundred of people exposed. Therefore, rounding may exceed by 250 people the agglomeration population (50 people per noise band).

When the population exposed exceeds 100% of the agglomeration population, we have adjusted the population exposed to the agglomeration population.

3.1.4 Visual inspection of the relationship between people exposed and number of inhabitants

The first step for conducting the regression is to look at the scatter plot between the two involved variables: people exposed and the number of inhabitants (Figure 3.3). At first sight, it seems to fit a perfect linear regression. However, given the skewed distribution of both variables, a log-log transformation shows a better approximation to a normal distribution of both variables (Figure 3.4).

In that case, we will test the following models:

People_exposed = a + b*Number_inhabitants

People_exposed = a + b*Number_inhabitants + c*Number_inhabitants^2

Log(People_exposed) = a + b*log(Number_inhabitants)

The second model corresponds to a polynomial regression of order 2, which is useful when there is a small bending on the relationship between the two variables. Although it is not evident that it would be useful in that case, we will include it as an illustration that it can be easily implemented and tested.

Zero values in the log transformation are problematic since log(0) is not a real number. Therefore, in the case of zero values, we should add a small amount to all zeros: 0,1. This does not impact the total number of people exposed while keeping the agglomeration in the regression analysis. This is relevant for L_{night}. As it is logical, no zero values have been observed in people exposed to L_{den} equal or greater to 55 dB.

Figure 3.3: Scatterplot of the number of inhabitants and people exposed to noise from roads inside agglomerations (L_{den} equal or greater than 55 dB). N = 329 agglomerations

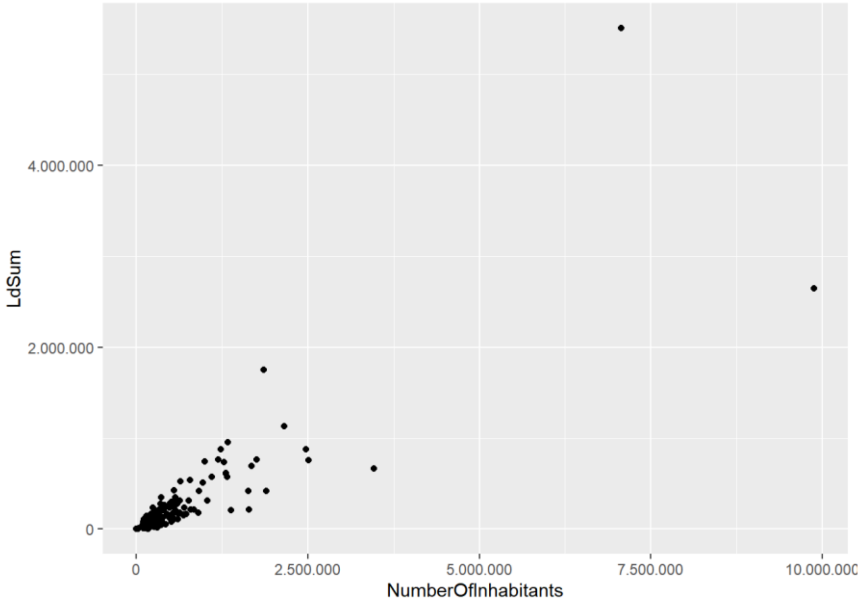
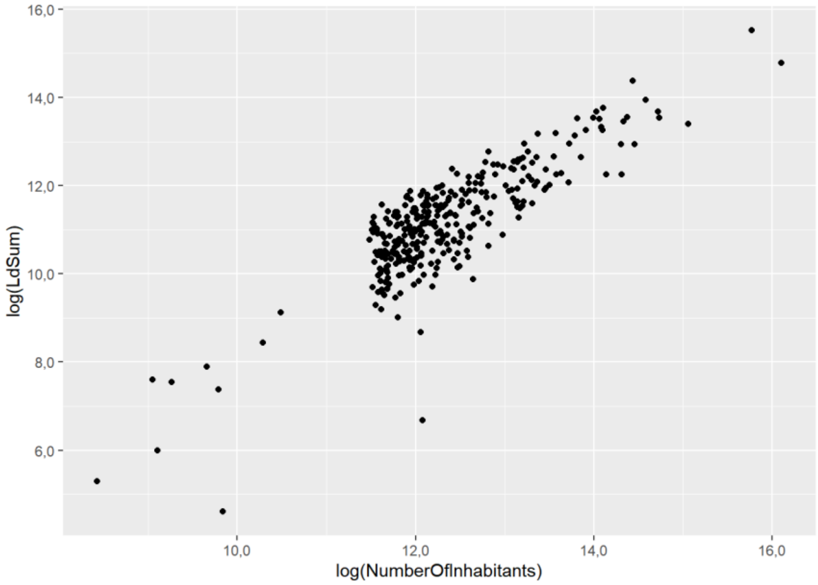


Figure 3.4: Log-log transformation of the number of inhabitants and people exposed to noise from roads inside agglomerations (L_{den} equal or greater than 55 dB). N = 329 agglomerations



3.1.5 Data subsetting for regression analysis and validation

Data subsetting refers to divide the agglomerations where data has been reported into two groups: one for estimating the regression parameters and the other one to validate the regression. Then, we apply the regression model to the second subset, validation, which is independent of the data used to estimate the model. Finally, the outcome can be compared with the original data.

The following requirements are needed for subsetting:

- The minimum number of samples (agglomerations). The total number of (complete) data should follow the rule (Snee, 1977)

N > 2*(nr of independent parameters) +25

In our case, we only include one parameter resulting in 27 as the minimum number of agglomerations required for a valid splitting.

- As a general rule, data is split by a 70:30 ratio, being 70 for estimating the model and 30 for validation (Snee, 1977).

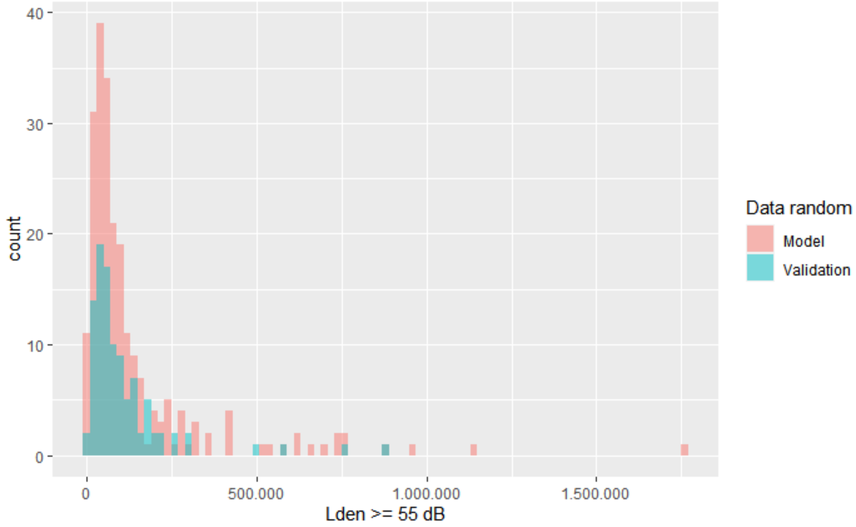
According to these rules, 226 agglomerations where data is reported -outliers excluded, have been randomly divided as follows:

- 226 agglomerations for estimating the regression model
- 103 agglomerations for validation

Both subsets must follow the same distribution. The Kolmogorov–Smirnov test is a nonparametric test of the equality of continuous one-dimensional probability distributions. The distribution of both model and validation subset are depicted in Figure 3.5. In that case, the probability of the Kolmogorov–Smirnov statistic is 0,47. Therefore, the null hypothesis that both data sets have the same distribution is not rejected.

When the distributions significantly differ, a new random subset needs to be selected until the condition of the same distribution is met.

Figure 3.5: Distribution of two subsets of agglomerations: agglomerations used to estimate the regression model and agglomerations used for validation



3.1.6 Test models

The three models described in step 2 have been calculated on the subset of 226 agglomerations selected for that purpose:

- Model 1. Linear. People_exposed = a + b*Number_inhabitants
- Model 2. Polynomic. People_exposed = a + b*Number_inhabitants + c*Number_inhabitants^2
- Model 3. Log-log. Log(People_exposed) = a + b*log(Number_inhabitants)

From the performance perspective, model 3 has the highest R^2, followed by model 2, and model 1 (Table 3.3). The other statistics related to the model accuracy can only be compared between model 1 and model 2 since are scale-dependent -model 3 has been log-transformed. In all cases, the lower of sigma, AIC, and BIC, the better. In that case, we see that model 2 has lower values than model 1.

Sigma measures the average error performed by the model in predicting the outcome (Table 3.3). Therefore, sigma could be read as the error on estimating the people exposed to noise: there is a small difference (about 6.300 people) between model 1 (229.022 people) and model 2 (222.724 people).

Finally, the F statistic p-value, which measures the statistical significance of the regression, is in line with the previous statistics: model 3 is more significant than model 2, and model 2 is more than model 1 (Table 3.3).

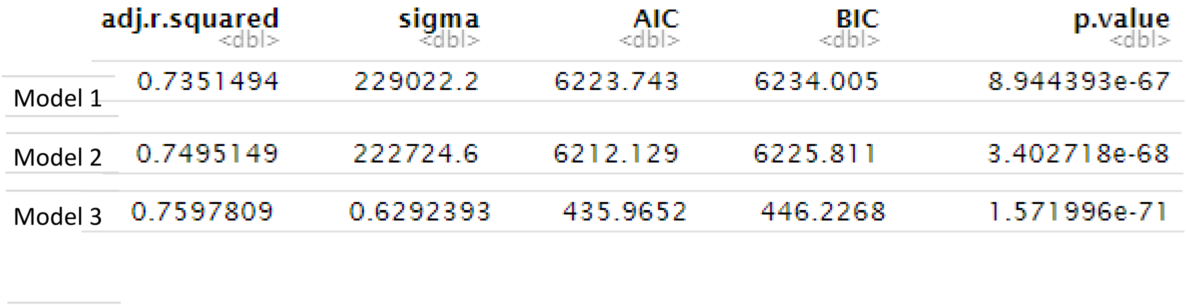
In addition to the accuracy, diagnostic plots are relevant to identify the possible weakness of the regression model. Figure 3.6 provides three of the most common diagnostic plots:

- Residual versus fitted. This plot shows if residuals have non-linear patterns. All models have a certain deviation: model 1 and model 2 have a strong deviation on agglomerations with higher exposure (right side of the figure). Model 3 has a smaller deviation on both extremes. This indicates that other factors may be relevant to predict people exposed, and the number of inhabitants is only one factor -probably the main factor given the high level of prediction.
- Normal Q-Q. This plot shows if residuals are normally distributed. Models 1 and 2 show that the extremes are problematic, while model 3 has a better fit to the line (normal distribution).
- Residuals versus leverage. This plot helps us to find influential cases, if any. Agglomerations that are outside the dotted red lines (Cook’s distance) are influential in the model, i.e. these

agglomerations have more weight on defining the model compared with the other agglomerations. Model 1 and Model 2 have three agglomerations with a strong impact on the model (three points outside the Cook's distance represented by the dotted line). The log transformation in model 3 was effective in removing the strong influence of extreme values.

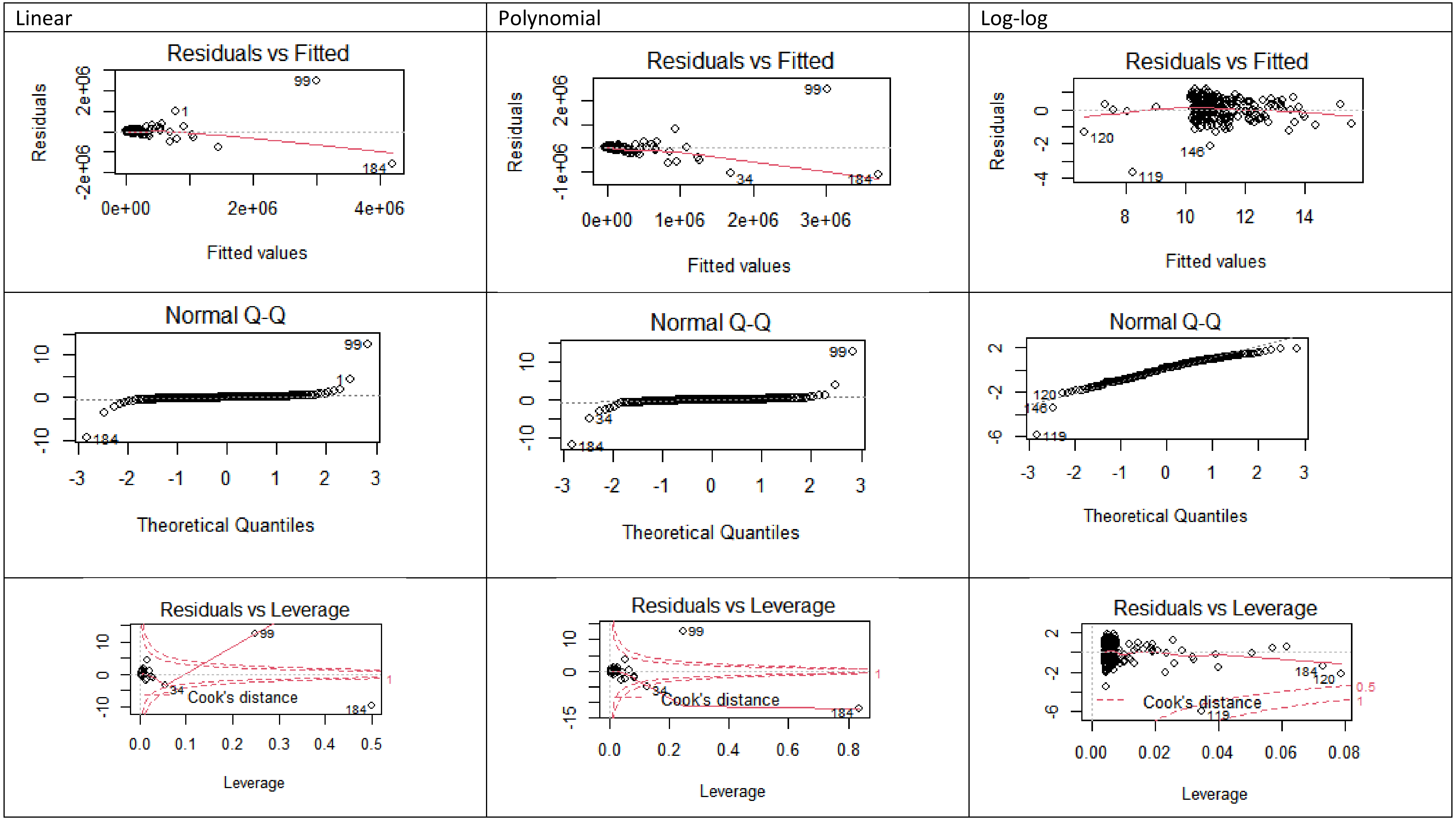
Details of the output are provided in Annex I.

Table 3.3: Statistics of the three tested regression models



	adj.r.squared	sigma	AIC	BIC	p.value
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
Model 1	0.7351494	229022.2	6223.743	6234.005	8.944393e-67
Model 2	0.7495149	222724.6	6212.129	6225.811	3.402718e-68
Model 3	0.7597809	0.6292393	435.9652	446.2268	1.571996e-71

Figure 3.6: Diagnostic plots for the three regression models: residuals versus fitted (first row), normal Q-Q (second row), and residuals versus leverage (third row). Number are identifiers of agglomerations. Red line, trend of the plot. Dotted lines, Cook's distance (0,5 and 1)



3.1.7 Validation

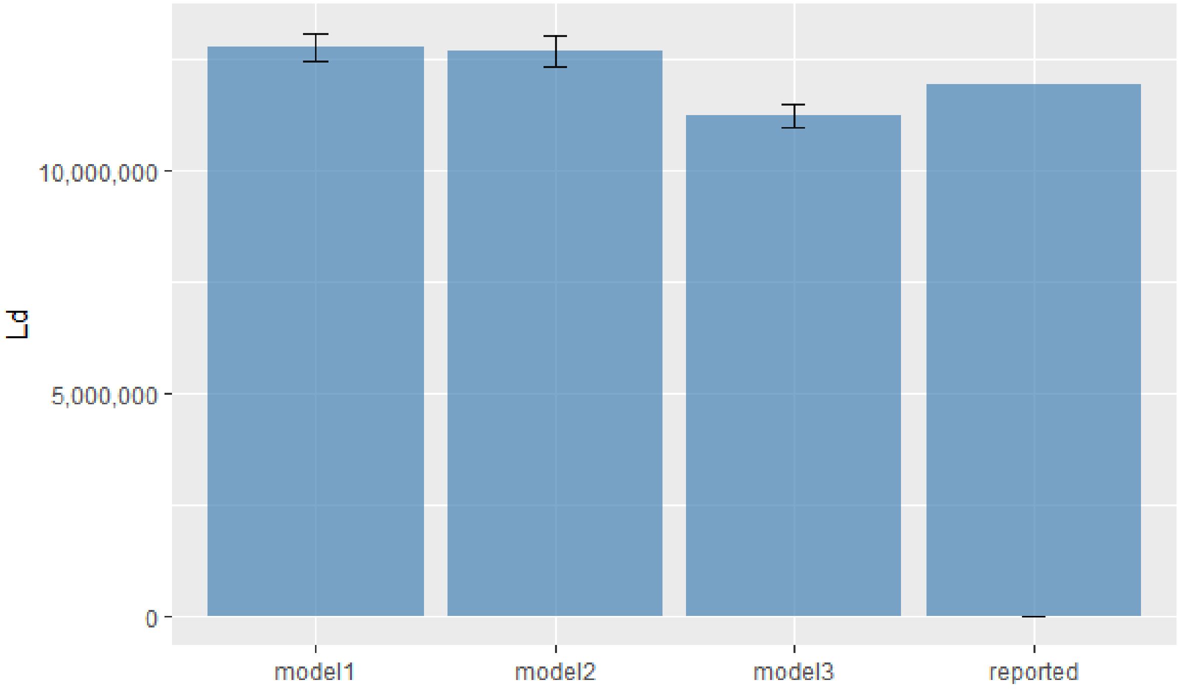
In the previous step, we have seen that model 3 looked better because of higher R^2 and better performance on the diagnostics.

Table 3.4 and Figure 3.7 show the results of applying the 3 regression models to the subset selected for validation (step 3). Model 1 and model 2 overestimate the population exposed, while model 3, closer to the reported values, underestimate the population exposed. The % of the difference between reported data and the estimates from the three models are very close, being model 3 being the one with a lower percentage (5,9%). Also, the confidence interval for model 3 is lower. Therefore, model 3 (log-log transformation) will be used to estimate the missing data. However, this result is data specific and could not be generalised. Therefore, the most appropriate regression model should be checked each time that the gap-filling is performed.

Table 3.4: Results of the validation of the three regression models. N = 103 agglomerations

	People exposed	% of difference	Confidence interval
Reported	11.951.291		
Model 1	12.777.297	6,9	323.394
Model 2	12.689.144	6,2	363.808
Model 3	11.245.629	-5,9	279.311

Figure 3.7: Validation of the three regression models to estimate people exposed to noise from roads inside agglomerations (L_{den} equal or greater than 55 dB). Confidence interval (95%) provided for the regression models. n = 103 agglomerations



3.1.8 Estimate values for missing data

Once the regression model has been selected, the following steps are taken:

1. For each agglomeration where the data is missing, estimate the population exposed by applying the regression model. In that particular case, since we selected a log-log regression we have to transform back the estimated population exposed (anti log).
2. Calculate the SE for each estimated value.

3.1.9 Calculate the total people exposed in Europe

1. Sum all the estimated values
2. We need to calculate the SE of the sum, which is obtained by quadrature of the individual SE
3. Finally, calculate the confidence interval.

3.1.10 Estimate the people distributed by noise bands

Once the estimated total number of people exposed is calculated (previous step), we distribute the population between the different noise bands.

1. Based on the total number of people exposed per each agglomeration reported by Member States, we calculate the percentage that each noise band represent versus the total number of people exposed, for Lden and for Lnight, and then we derive the mean at European level. It needs to be taken into consideration that the percentage values have been obtained discarding the agglomerations providing 0 people exposed in all noise bands (L_{den} and L_{night}, or L_{den}, or L_{night}). Due to the rounding process, 0 could mean 0 to 49 people exposed; therefore, multiple combinations are possible with the same outcome of 0 people exposed. Consequently, an equal attribution of 20% of people exposed to each noise band only represents one of the multiple possible combinations. For that reason, agglomerations with 0 people exposed are excluded.
2. We apply the percentages to the agglomerations where we have estimated the total population, with the corresponding error of the estimate.
3. Finally, we aggregate all the agglomerations at European level, with the corresponding estimation of the confidence interval.

3.2 Gap filling method for railway noise, aircraft noise and industrial noise inside agglomerations

This section summarises the method applied to gap fill exposure information for railways noise, aircraft noise and industrial noise inside agglomerations.

Figure 3.8 provides an overview of the process for estimating missing data, as Ramos (2019) described.

In each step, the decision tree uses the best approach according to the available ancillary data:

- Use data from the previous reporting period if available.
- When data from the previous reporting cycle is not available, data is estimated with the European average of the % of population exposed inside the agglomeration. In that case, no significant correlation was found between population exposed and other predictor parameters (e.g., the agglomeration population, area of the agglomeration); therefore, the European average is the best alternative (Fons et al., 2015).
- Based on the total number of people exposed per each agglomeration reported by the Member States, we calculate the percentage that each noise band represent versus the total number of people exposed, for L_{den} and for L_{night}. Then we derive the mean at European level. It needs to be taken into consideration that the percentage values have been obtained discarding the agglomerations providing 0 people exposed in all noise bands (L_{den} and L_{night}, or L_{den}, or L_{night}). Due to the rounding process, 0 includes figures ranging from 0 to 49 people exposed; therefore, multiple combinations are possible with the same outcome of 0 people exposed. Consequently, an equal attribution of 20% of people exposed to each noise band only represents one of the several possible combinations. For that reason, agglomerations with 0 people exposed are excluded.
- We apply each noise band's percentages to the agglomerations where we have estimated the total population, with the corresponding error of the estimate.
- Finally, we aggregate all the agglomerations at European level, with the corresponding estimation of the confidence interval.

A more detailed explanation and the basis for the methodology applied in each step is provided in Table 3.5.

The gap filling methodology also includes those few cases where data is missing for some noise bands. Then, **partial gap filling** is conducted as follows:

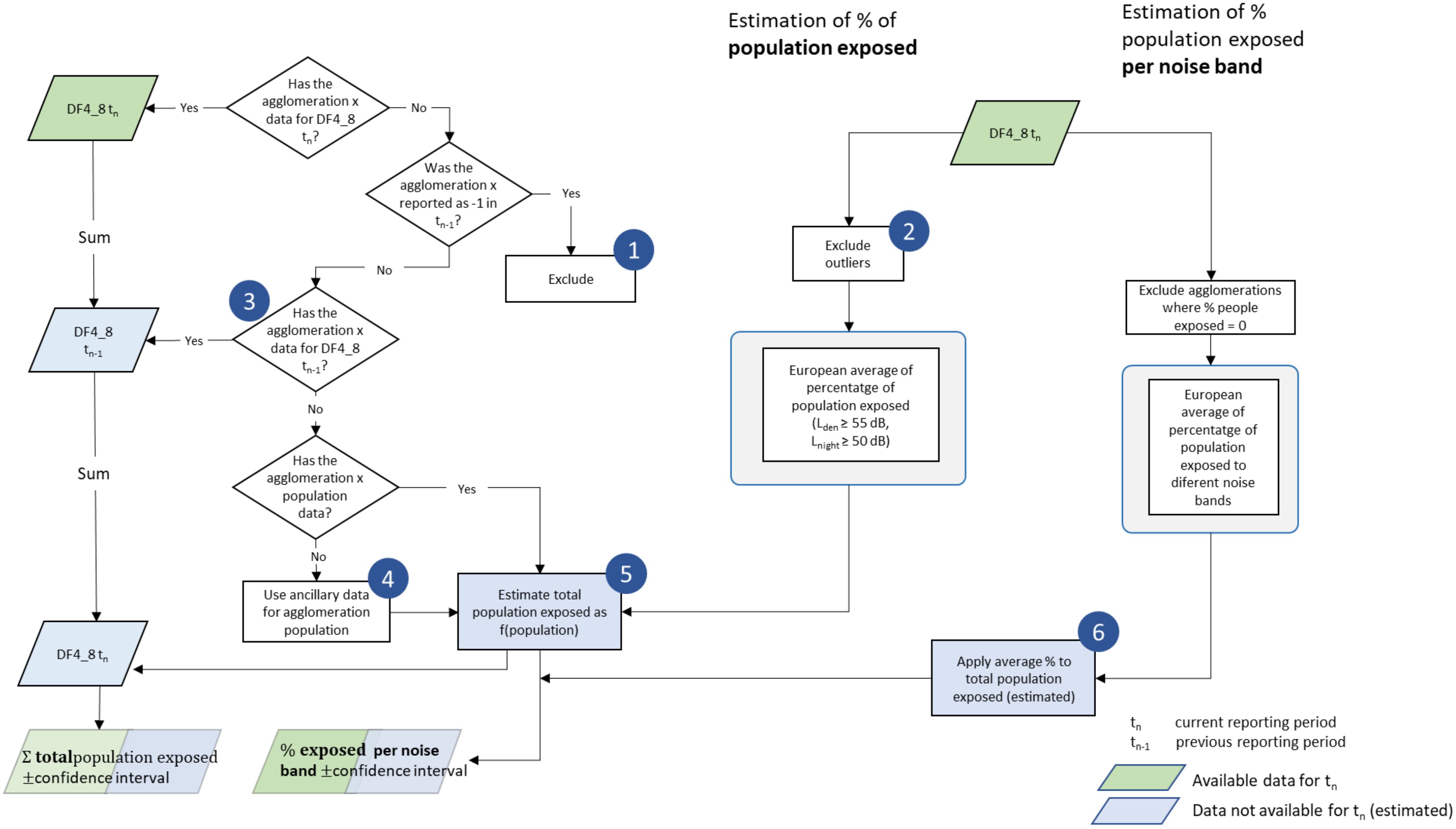
- Use data from the previous reporting period for the missing noise band(s).
- If data from the previous reporting period is not available use the European average of % of the population exposed by noise band to estimate the missing data.
- Table 3.2 provides an example of a partial gap filling when data from the previous period is not available.

Each step that involves the estimation of data, corresponding error is calculated. Finally, these errors are propagated to the aggregated European figures as described in sections 3.1.9 and 3.1.10.

Table 3.5: Methods used on the different steps of gap filling people exposed to noise from railways, airports and industry inside agglomerations. Numbers refer to the steps highlighted in Figure 3.1

Step	Method for gap filling	Explanation	Comment
1	Exclude the agglomeration if it was reported as -1 (not applicable) at t_{n-1}	An agglomeration reporting -1 for a source in the previous reporting cycle (t_{n-1}), meant that that source was not applicable according to the END specifications. Therefore, we retain the non-applicability at t_n (Fons et al., 2016). Check done in previous gap-filled data (2016, 2017, 2018 and 2019) demonstrates that the assumption was correct in 90% of cases. Therefore, the potential underestimation of the European figure (data excluded) is more accurate than the overestimation, when all these agglomerations are gap filled.	This step only applies if data is not reported, and "not applicable" is not explicitly mentioned at t_n.
2	Exclude outliers from the reported data to be used for the European average (step 5)	Outliers from the change percentage between t_{n-1} and t_n based on the interquartile range (IQR, the difference between 3rd and 1st quartile). The exact threshold is ± 1,5 IQR.	Previous assessments have identified some extreme population changes exposed to noise between two reporting periods (Ramos, 2019). Therefore, the estimation of missing data in 2019 (Ramos, 2019), adopted a methodology to exclude outliers.
3	Use the data delivered in the previous reporting period for the same agglomeration, if available.	A comparative analysis described in Fons et al. (2017) concluded that this is the method with the smaller error.	The method is constrained to the availability of the data on the previous reporting cycle.
4	Estimate the total population from ancillary data.	If the country reported the agglomeration's delineation, the population could be derived from Urban Atlas described in Fons et al. (2015). The average error of the estimation based on Urban Atlas is 3%, ranging from 1 to 10%. If the delineation has not been reported, data from Eurostat can be used. The agglomeration population is the used in step 5.	The reporting cycle of the Urban Atlas is always one year later than the END.
5	Estimate the population exposed by multiplying the population of the agglomeration with the European average of the % of people exposed.	No significant correlation was found between population exposed and other predictor parameters (e.g., the agglomeration population, area of the agglomeration); therefore, the European average is the best alternative (Fons et al, 2015).	
6	Estimate the % of the population exposed per noise bands (%) from the European average.	Based on the total number of people exposed per each reported agglomeration, we calculate the percentage that each noise band represent versus the total number of people exposed, for L_{den} and for L_{night}. Then we derive the mean at European level. This approach is discussed in depth in Fons et al. (2015).	Initially, the % was calculated on a country basis when were enough agglomerations reported. Fons et al. (2017) concluded that using the country average has a similar error than the European average. Therefore, the European average is used for simplicity.

Figure 3.8: Overview of the process for estimating the population exposed to noise from rails, airports or industry inside agglomerations when data is not available. The methodology only applies to agglomerations that have to report according to END requirements. Numbers indicate specific methods for gap filling depending on available ancillary data -details are described in Table 3.4. Source: updated from Ramos (2019)


Estimation of % of
population exposed
Estimation of % population exposed
per noise band
DF4_8 t_n
Yes
Has the agglomeration x data for DF4_8 t_n?
No
Was the agglomeration x reported as -1 in t_{n-1}?
Yes
1
Exclude
No
Sum
3
DF4_8 t_{n-1}
Yes
Has the agglomeration x data for DF4_8 t_{n-1}?
No
Has the agglomeration x population data?
Yes
Sum
No
4
Use ancillary data for agglomeration population
5
Estimate total population exposed as f(population)
DF4_8 t_n
Σ total population exposed ±confidence interval
% exposed per noise band ±confidence interval
DF4_8 t_n
2
Exclude outliers
European average of percentatge of population exposed ($L_{den} \geq 55$ dB, $L_{night} \geq 50$ dB)
Exclude agglomerations where % people exposed = 0
European average of percentatge of population exposed to diferent noise bands
6
Apply average % to total population exposed (estimated)
t_n current reporting period
t_{n-1} previous reporting period
Available data for t_n
Data not available for t_n (estimated)

4 END major roads and major railways exposure data outside agglomerations: gap filling

4.1 Gap filling method

This method is based on the working paper establishing a methodological proposal to interpolate a complete coverage on noise exposure at EU level (ETC/ACM, 2015).

Figure 4.1 provides an overview of the process for estimating missing data, as Ramos (2019) described.

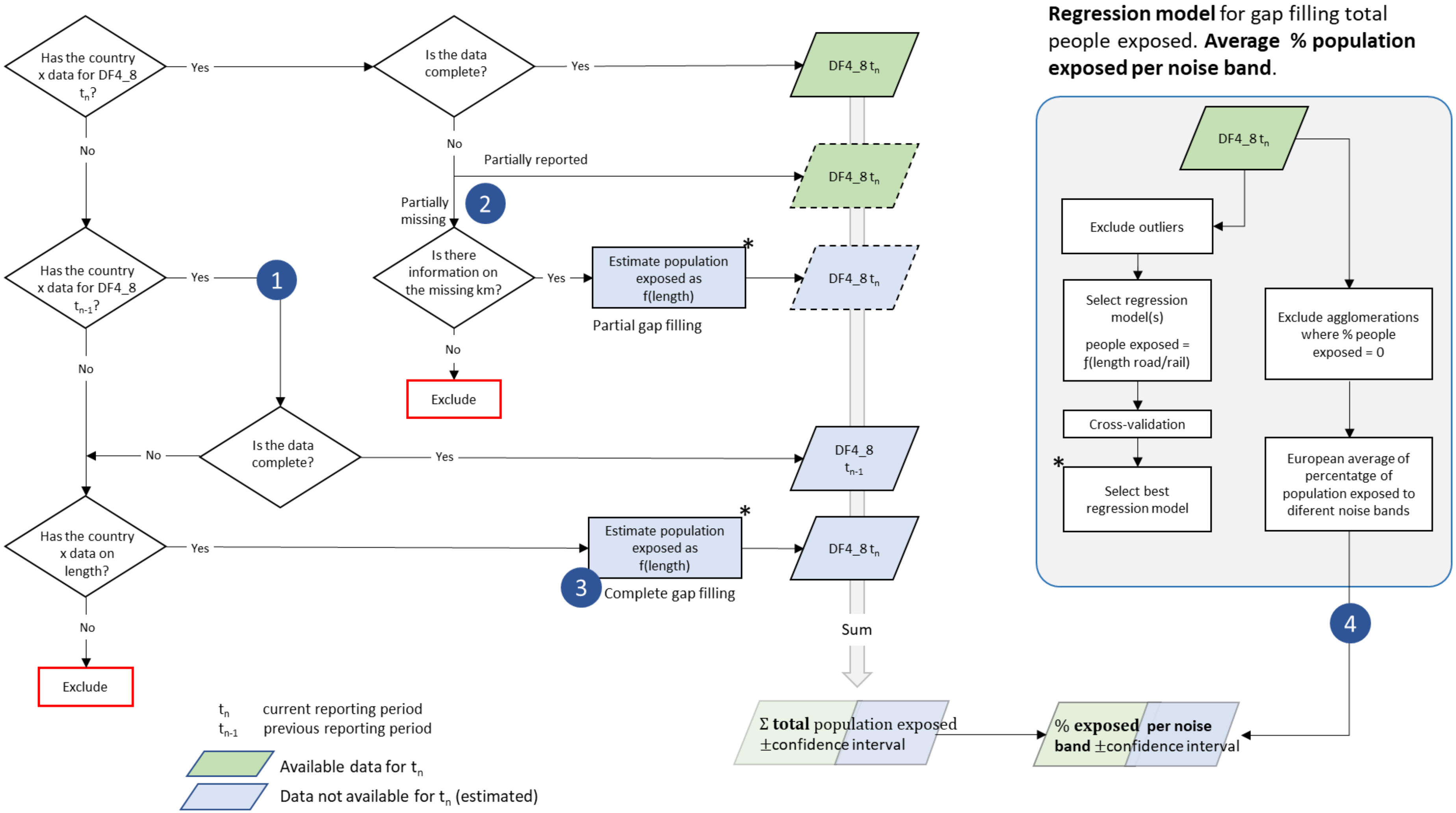
In each step, the decision tree uses the best approach according to the available ancillary data:

- Partial gap filling if reported data is not complete. Data completeness can only be evaluated if exposure has been delivered by the road and rail segments. Then network segments are linked to DF1_5 dataflow to match segments to be reported with the actual data reported. Missing segments are gap filled with the regression between people exposed and the length of the transport network (the procedures is the same as described for roads inside agglomerations. It should be noted that the length of the transport network also includes major source inside agglomerations. However, the data reported on people exposed refers only to people outside agglomerations. When the exposure information has been delivered as one single value for the entire network, the codes are supplied as -1 or -2 or the codes between dataflows (DF1_5 and DF4_8) do not match, then the comparison of the code is not possible, and the dataset is assumed as complete.
- If data is not reported, use data from the previous reporting period if available and complete (same procedure as explained in the above bullet point to evaluate completeness).
- When data from the previous reporting cycle is not available or not complete, the regression between the number of people exposed and the transport network's length has been calculated. Then the regression has been applied to estimate missing data. In that case, complete gap filling is applied, i.e. even if some data (incomplete) is available from the previous period, the regression is applied to the full extent of the transport network. It should be noted that the length of the transport network also includes major source inside agglomerations. However, the data reported on people exposed refers only to people outside agglomerations.
- Once the total number of people exposed is estimated, we distribute the total population exposed to the different noise bands based on the European average of the population exposed by noise bands. The European average is calculated with all the available data, even if it is incomplete for a certain region of the country. The European average discards the countries or regions providing 0 people exposed in all noise bands. Due to the rounding process, 0 could mean 0 to 49 people exposed. Therefore attributing 20% to each noise band would not be accurate since other options would also be feasible.

Table 4.1: Methods used on the different steps of gap filling people exposed to noise from roads inside agglomerations. Numbers refer to the steps highlighted in Figure 3.1

.Step	Method for gap filling	Explanation	Comment
1	Use the data delivered in the previous reporting period, if available.	A comparative analysis described in Fons et al. (2017) concluded that this is the method with the smaller error.	The method is constrained to the availability of the data on the previous reporting cycle, and the data is complete.
2	When the data is not complete for a certain country, estimate the missing data with the regression between people exposed and the length of the transport network.	The methodology for partial gap filling is described in Fons et al. (2015). The regression and correlation analysis between population exposed and potential predictors (country area, length of transport network) is documented in Fons et al. (2015). Later on, Fons et al. 2017) demonstrated that, if available, using data from the previous reporting period (step 1) is more accurate than the regression approach.	The current report provides a detailed description of the metrics to evaluate the best regression model, including estimating the error and confidence interval in the final aggregation of data (EU figures). There is no a priory regression model to be applied each time that the gap filling is developed. The regression model needs to be checked each time since the relationship is strongly dependent on the data included for the regression. The approach described in section 3.1 is also valid here.
3	Estimate the population exposed from the regression between population exposed and km of road or rail.	The regression and correlation analysis between population exposed and potential predictors (country area, length of transport network) is documented in Fons et al. (2015). Later on, Fons et al. 2017) demonstrated that, if available, using data from the previous reporting period (step 1) is more accurate than the regression approach.	The current report provides a detailed description of the metrics to evaluate the best regression model, including estimating the error and confidence interval in the final aggregation of data (EU figures). There is no a priory regression model to be applied each time that the gap filling is developed. The regression model needs to be checked each time since the relationship is strongly dependent on the data included for the regression. The approach described in section 3.1 is also valid here.
4	Estimate the % of the population exposed per noise bands (%) from the European average	Based on the total number of people exposed, we calculate the percentage that each noise band represent versus the total number of people exposed, for L_{den} and for L_{night}, and then we derive the mean at European level. This approach is discussed in depth in Fons et al. (2015).	

Figure 4.1: Overview of the process for estimating the population exposed to noise from major roads and major rails when data is not available. The methodology only applies to countries that have to report according to END requirements. Numbers indicate specific methods for gap filling depending on available ancillary data -details are described in Table 4.1. Dotted lines: data partially reported, and data partially gap filled. Source: updated from Ramos (2019)


Has the country x data for DF4_8 t_n?
Yes
Is the data complete?
Yes
DF4_8 t_n
No
Partially reported
DF4_8 t_n
No
Partially missing
2
Is there information on the missing km?
Yes
Estimate population exposed as f(length)
*
DF4_8 t_n
Partial gap filling
No
Exclude
Has the country x data for DF4_8 t_{n-1}?
Yes
1
No
Is the data complete?
No
Yes
DF4_8 t_{n-1}
Has the country x data on length?
Yes
Estimate population exposed as f(length)
*
DF4_8 t_n
3
Complete gap filling
No
Exclude
Sum
Σ total population exposed ±confidence interval
% exposed per noise band ±confidence interval
t_n current reporting period
t_{n-1} previous reporting period
Available data for t_n
Data not available for t_n (estimated)
Regression model for gap filling total people exposed. **Average % population exposed per noise band**.
DF4_8 t_n
Exclude outliers
Select regression model(s)
people exposed = f(length road/rail)
Cross-validation
*
Select best regression model
Exclude agglomerations where % people exposed = 0
European average of percentatge of population exposed to diferent noise bands
4

5 END Major airport exposure outside agglomerations: gap filling

5.1 Overview

The current methodology to estimate missing data on the population exposed to major airports outside agglomerations is presented in Figure 5.1 (Ramos, 2019). In this case, we start with the calculation of the average relative change of population exposed between the current reporting period (t_2) and previous cycle (t_1) as described in Jones (2013):

$$Relative\ difference = \frac{\sum_{i=1}^{n}\left(\frac{People\ exposed_{i_{t2}} - Pople\ exposed_{i_{t1}}}{People\ exposed_{i_{t1}}}\right)}{n}$$

Where *n* is the number of major airports with reported data for the period t_1 and t_2, being t_2 the current reporting cycle, and *i* is the *i*th major airport where data is available.

In practical terms this relative difference can be expressed as ratio and directly applied to those major airports where data for the current reporting period is missing, but available for the previous period:

$$Ratio = \frac{\sum_{i=1}^{n}\left(\frac{People\ exposed_{i_{t2}}}{People\ exposed_{i_{t1}}}\right)}{n}$$

Then, this ratio It should be noted that the use of the ratio of change could be extended back up to two reporting cycles. For example, in the case that data for a certain major airport is only available for 2007, then we calculate the ratio of change for the period 2007 – 2017 and apply this ratio to the data of 2007.

In this case outliers have also to be excluded from the calculation as explained in the case of roads inside agglomerations.

The use of the relative difference is based on the fact that it provides better estimates than any other predictor, e.g. number of annual flights (Fons et al., 2016). The low correlation between people exposed and the number of flights is explained by the fact that exposure to aircraft noise is strongly dependent on local conditions, including specific operational measures (take-off and landing routes, time of the day,…), meteorological conditions, or land use planning. However, the current approach has its limitations since it assumes a homogenous change ratio in all agglomerations. Figure 5.2 and Figure 5.3 illustrate the distribution of the relative difference between 2012 and 2017. The value ranges from -1 (100% decrease on exposure) to 1,5 (150% increase of population exposed). About 9 out of 40 major airports are outliers (23%).

Figure 5.1: Overview of the workflow for estimating the population exposed to major airports outside agglomerations when data is not available. Source: Ramos (2019)

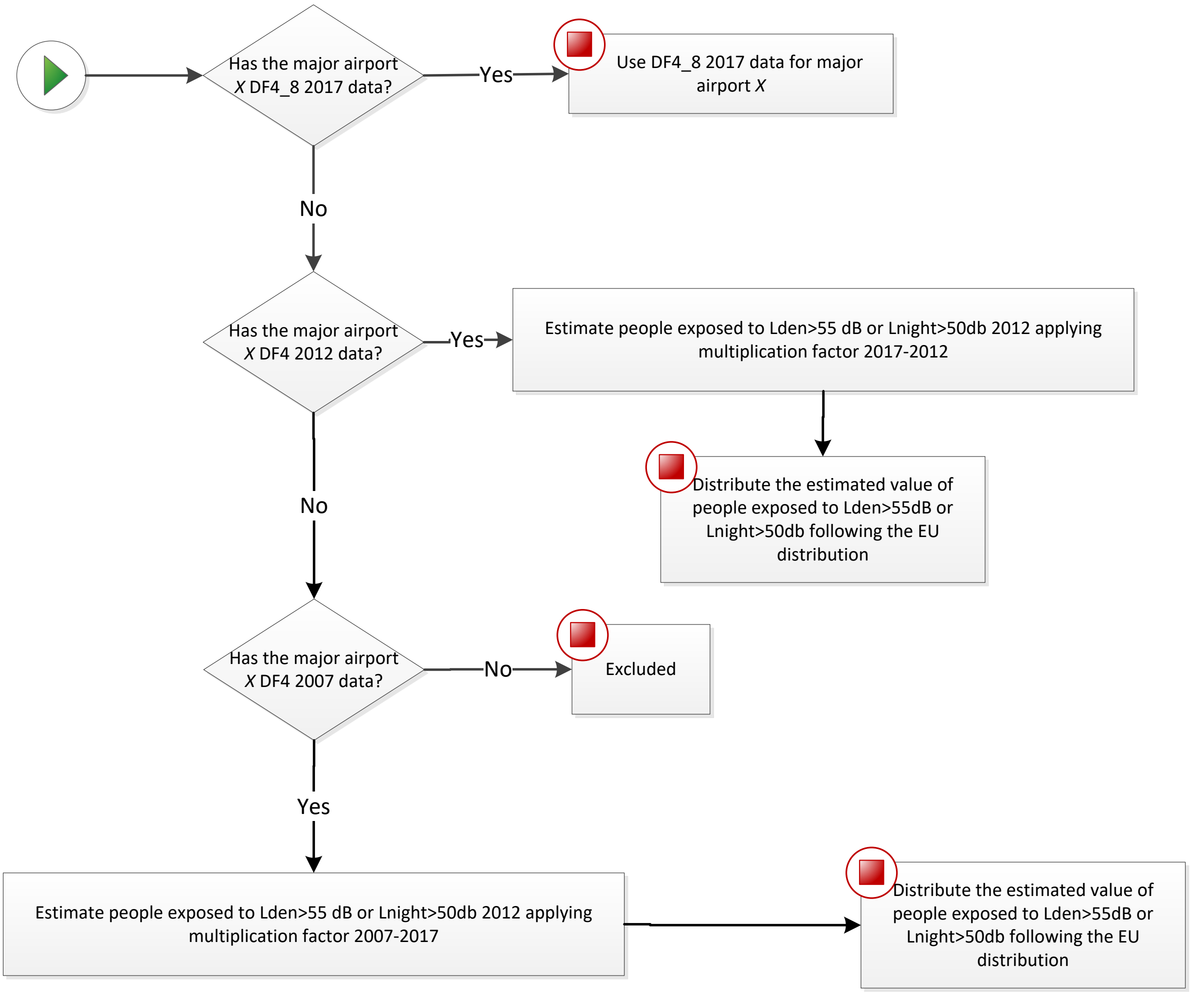

Has the major airport X DF4_8 2017 data?
Yes
Use DF4_8 2017 data for major airport X
No
Has the major airport X DF4 2012 data?
Yes
Estimate people exposed to Lden>55 dB or Lnight>50db 2012 applying multiplication factor 2017-2012
Distribute the estimated value of people exposed to Lden>55dB or Lnight>50db following the EU distribution
No
Has the major airport X DF4 2007 data?
No
Excluded
Yes
Estimate people exposed to Lden>55 dB or Lnight>50db 2012 applying multiplication factor 2007-2017
Distribute the estimated value of people exposed to Lden>55dB or Lnight>50db following the EU distribution

Figure 5.2: Box plot of the relative change of population exposed to noise from major airports between 2012 and 2017 (L_{den} ≥ 55 dB). Outliers are indicated in red. The lower and higher limits of the box correspond to the 1st and 3rd quartile, respectively. The horizontal line inside the box is the median

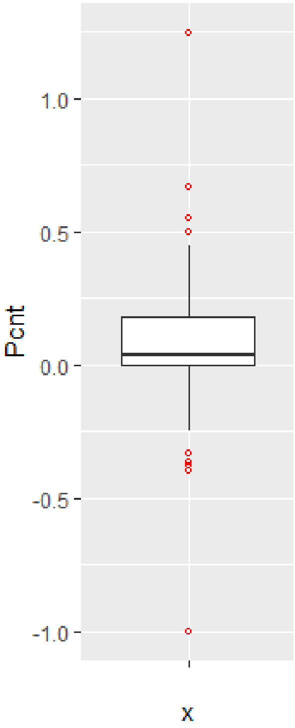
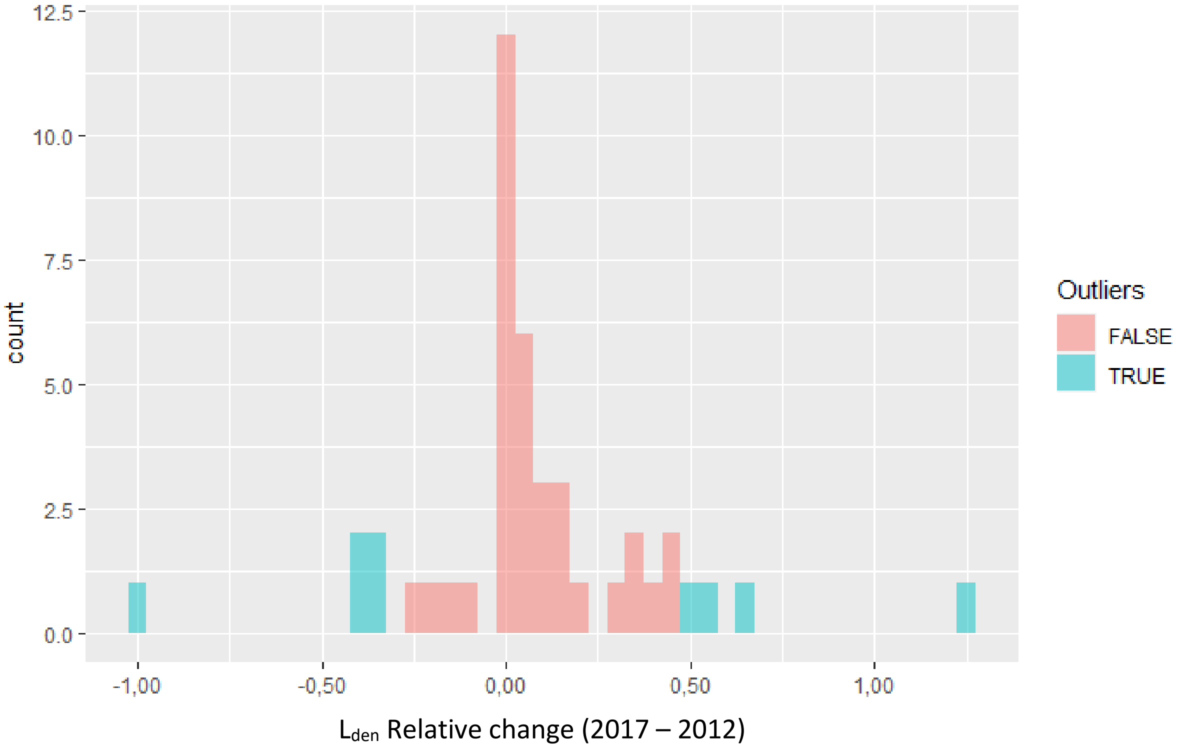


Figure 5.3: Distribution of the population relative change exposed to noise from major airports between 2012 and 2017 (L_{den} ≥ 55 dB). Outliers in green



Since the outliers are equally distributed around the mean, there is no major impact on the average ratio if we include them in the calculation (Table 5.1). However, the standard error decreases by 40% when outliers are excluded. Moreover, if outliers are not excluded, the estimated confidence interval of the change ratio ranges from a small decrease (-0,03) to 0,18 increase because the standard error (0,10) is higher than the average (0,08), as presented in Table 5.1.

Table 5.1: Mean, upper and lower boundaries of the confidence interval (95%), and standard error (SE) of the change of population exposed to noise from major airports between 2012 and 2017 ($L_{den} \geq 55$ dB). Data is presented for the complete set of available major airports and for the subset without outliers. N is the number of major airports

	Lower boundary	**Average**	**Upper boundary**	**SE**	***n***
With outliers	-0,03	0,08	0,18	0,10	40
Without outliers	0,03	0,08	0,14	0,06	31

Given all these uncertainties, the current report provides an alternative method and test its suitability and potentially higher performance. Our hypothesis is that delineating an approximated noise contour band in those major airports that have not reported data will better estimate the population exposed than the current method. We assume that with this approach, we better capture local conditions with a reasonable effort of computation.

The methodology has been tested in 10 airports where all the information has been reported (Table 5.2).

Table 5.2: Airports selected to test the proposed methodology. Airports with the same arrangement of runaways have the same colour

Airport	**ICAO code**	**Annual traffic**	**Runaways**
Berlin-Tegel	EDDT	182200	2 runways (side-by-side)
Berlin-Schönefeld	EDDB	70324	2 parallel runways
Copenhagen	EKCH	251799	3 runways cross model
Hamburg	EDDH	153876	2 runways cross model
Helsinki-Vantaa	EFHK	168704	3 runways, 2 parallel, one crossing (almost perpendicular)
Lisbon	LPPT	159795	2 runways cross diagonal disposal (45 degrees)
Milano-Malpensa	LIMC	166509	2 parallel runways
Napoli	LIRN	64712	1 runway
Wien	LOWW	226811	2 runways diagonal disposal
Budapest Ferihegy	LHBP	96705	2 runways diagonal disposal

5.2 A methodology based on estimating the noise contour around runaways

5.2.1 Overview

The methodology to estimate the population exposed to major airports (L_{den}) is synthesised in Table 5.3. The method for L_{night} would be similar.

As can be seen, the methodology has two major elements of uncertainty:

- Delineation of the noise contour around the major airports. These contours depend on several factors: number and length of runaways, local regulations, meteorological conditions, or orography, just to name some of them.
- Assuming that all the population living inside the contour is exposed to noise. Therefore, existing measures like building insulation are not considered since it is not feasible to introduce this component at European scale.

Table 5.3: Methodology to estimate population exposed to major airports (inside and outside agglomerations)

Steps	Output	Comments
1. Delineation of the noise contour for L_{den} 55dB (lower boundary)	Noise contour for L_{den} 55dB (lower boundary)	
a. Identify runaway(s)		From satellite imagery draw a simple line representing the full length of each runaway
b. Delineation of the contour for L_{den} 55dB around runaways		Delineation of the contour based on a certain buffer around the runaways
2. Cross the contour of L_{den} 55dB with the delineation of the agglomeration	In case of the presence of one or more agglomerations: contour outside and contour inside agglomerations(s)	This step is needed to differentiate the people exposed **outside** and **inside** the agglomeration (if the contour intersects with one or more agglomerations)
3. Calculate the population inside the noise contour (inside and outside the agglomeration)	Distribution of the peoples exposed by noise bands	Cross the areas of the previous step with the population grid. The obtained value will be an estimate of the population exposed to a major airport (inside and outside agglomerations).
4. Distribute the total population exposed to major airport by noise bands		Apply the European average of % of population exposed distributed by noise bands.

5.2.2 Data requirements

The following data has been used to test the methodology:

- Reference image for runaways: Google Maps
- Delineation of agglomerations as provided by the information reported by countries according to END specifications
- Noise contour bands reported by countries
- Population. Outside agglomerations, GEOSTAT population at 1 km grid[2]

5.2.3 Delineation of the runaways and noise contours

The most critical issue is the delineation of the noise contour. During the testing phase, it was taken into consideration the use of wind maps and its monthly/year direction means as usually aircraft depart and arrive counter wind. As well ATS routes, traffic maps and the use of waypoints and fixes to determine routes were analysed to determine the most common path of planes in each airport, but unfortunately, no relevant data was found. Many airports implement noise mitigation techniques (noise abatement procedures), that increases the uncertainty when creating a common delineation noise model, including the following:

- Defining noise abatement procedures that avoid residential areas as far as possible and avoid over-flying sensitive sites such as hospitals and schools
- Using continuous descent approaches and departure noise abatement techniques
- Ensuring that the optimum runway(s) and routes are used as far as conditions allow
- Avoiding unnecessary use of auxiliary power units by aircraft on-stand
- Building barriers and engine test-pens to contain and deflect noise
- Towing aircraft instead of using jet engines to taxi
- Limiting night operations
- Limiting the number of operations or the extent of a critical noise contour
- Providing noise insulation for the most severely affected houses
- Applying different operational charges based on the noisiness of the aircraft
- Monitoring individual noise levels and track keeping and penalising any breach

However, since this approach is quite effort consuming, it was decided to take one airport as a model and replicate the noise contour on the airports without data, adapted to the length of the runaways. For the one runway setup airports, the replication of the Tegel delineation is enough to have a strong approach close to the reality on the gap filled airports. When talking about two or three-runway setup, the complexity is higher. Despite that, taking into account what has been reported on contour maps, Table 5.4 shows the type of artificial delineations created to calculate the population exposed and gap-fill missing data:

In relation to the generated contour maps for gap filling, the best way to replicate the possible track of the aircraft is to look at what is happening in the airports; there is information reported. Depending on many factors such as frequent wind flows, orography, local regulations, etc, an aircraft creates one type of noise path or another. On perfect conditions, the most common is that the plane creates a

[2] https://ec.europa.eu/eurostat/web/gisco/geodata/reference-data/population-distribution-demography/geostat

noise track similar to what was reported in the Berlin-Tegel airport, a very smooth pattern of noise where small bellies, corresponding to the end of the runway, where aircraft reaches its maximum power in corresponding to the end of the runway, where aircraft reaches its maximum power in the ground, arises in the middle of this ellipse form.

For the one runway setup airports, the replication of the Tegel delineation is enough to have a strong approach close to the reality on the gap filled airports. When talking about two or three-runway setup, the complexity is higher. Despite that, taking into account what have been reported on contour maps the following table shows the type of artificial delineations created to calculate the population exposed and gap-fill missing data.

Table 5.4: Approaches to estimate the noise contour band accordign to type of runaway

Type of runway	Delineation model	Rationale	Approaches
1 runway	Tegel ellipse	the most common type of aircraft noise track	for shorter runways scale the Tegel ellipse the proportional difference of the runway length in relation with the Tegel main runway length (3km)
2 runways cross	Tegel ellipse both runways	hard to determine which of the two runways have more movements than the other producing larger noise tracks. In general main runways are built on an east-west line and the secondary track to fit local conditions of winds, orography, etc. The solution found was to decrease Tegel ellipse to half of its area in the “secondary” (not main) runway oriented north-south.	The secondary approach is to use the length of the runway as a factor to determine the size of the Tegel ellipse. Check Wien case study which has 3,5km each runway and Lisbon having the main runway with 3,8km, and the secondary runway with 2,4km. Precise runway length available under external links (3)
2 runways parallel	Tegel ellipse both runways	normally very similar noise behaviour for both runways.	Tegel ellipse seems to fit most of the cases when replicated twice in this case. Use the length of runways as an indicator of the Tegel ellipse size. Scale accordingly.
3 runways, 2 parallel, 1 crossing	Tegel ellipse parallel runways and 1/3 size Tegel ellipse for crossing runway	In most cases, the third runway is shorter having less movements being used mainly for local flights or when weather conditions are rough. The proposal is to reduce the Tegel ellipse to 1/3 of its area approximately. In general very few movements/year are produced in that runway.	Use the length of runways as an indicator of the Tegel ellipse size. Scale accordingly.
3 runways (or more) other setup	Tegel ellipse parallel runways and 1/3 size Tegel ellipse for crossing runway	The approach in this case is to use Tegel ellipse for the larger runways and decrease the ellipse to 1/3 if the runway is smaller.	Use the length of runways as an indicator of the Tegel ellipse size. Scale accordingly.

5.2.4 Detailed description of the methodology

The methodology could d be described as follows:

- copy model contour map delineation into a new feature
- move the central vertex into the central part of the runway and replicate it to all the airports being analysed
- rotate the delineation so it fits the orientation of the runway
- scale delineation into 1/3 of the model for secondary runways generally with less traffic
- dissolve delineations into one single area. Only in the case of more than 1 runway
- calculate area by hectares for all airports and all areas, reported contours and generated contours for gap filling
- use identity to print the agglomeration border into the model delineation so the areas are separated into inside or outside agglomeration
- to calculate the population exposed, use "extract by mask" tool and select the working zone as the mask and the population raster as the data to extract
- convert the output raster of the population exposed into integer values with the "int" tool
- sum the values of the population for each specific raster zone

5.2.5 Outcome of the test

Table 5.5 provides an overview of the result of two methodologies to estimate people exposed to noise from major airports when this information has not been reported:

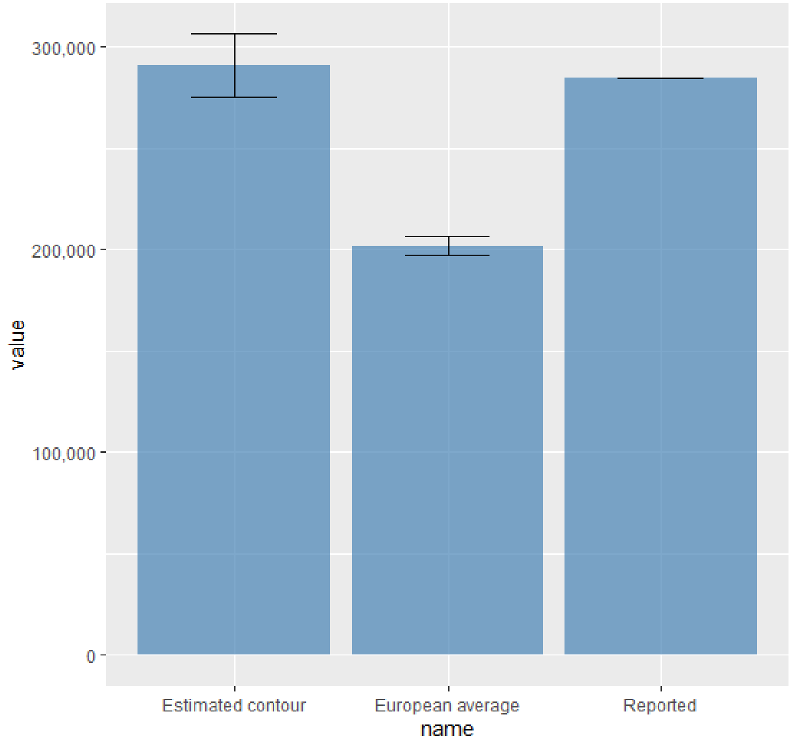
- Using a European average with the reported data
- Delineating the noise contour for L_{den} 55 dB, and calculate the population inside.

As can be seen, the gap-filling with the European average is less accurate but has higher precision, while the outcome of delineating the noise contour is the opposite: it is very accurate, although three times less precise. The implication on the final values can also be seen in Figure 5.4. In that case, it seems that is preferable a more accurate estimation (closer to the reality) with a reasonable precision of 7%, which means that the real value will be ± 20.000 people.

Table 5.5: Accuracy and precision of the two tested methodologies to the gap-fill population exposed to major airports outside agglomerations. Precision values provided as the number of people exposed are illustrative and valid for a reported value of 284.000 people exposed (9 major airports)

Methodology	Description	Accuracy	Precision	
			%	People exposed
European average	Very easy to apply. Just calculate the average ratio of change between two periods	29%	2,2%	4.500
Delineation of the contour for L_{den} 55 dB	Calculation intensive. Needs to identify the runaways, delineate the contour, exclude the area inside the agglomeration, and attribute the population.	2,3%	6,9%	20.000

Figure 5.4: Reported and estimated values for people exposed to noise from 9 major airports (L_{den} ≥ 55 dB). Reported, reported data. Estimated contour, people exposed to noise has been estimated by delineating an approximated contour band. European average, people exposed has been estimated by applying to the data from the previous reporting cycle the European average of change of population exposed



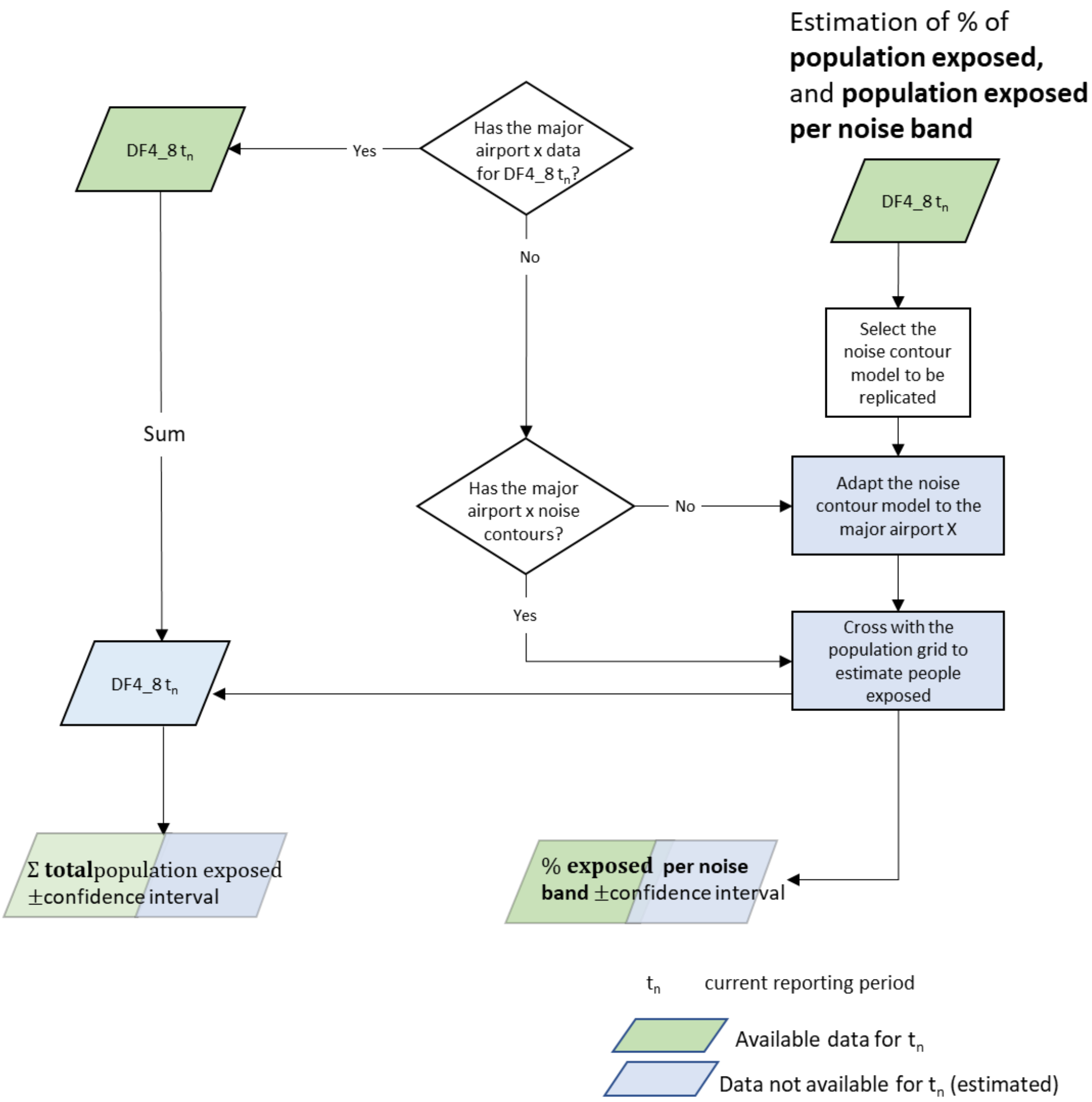
5.3 Summary of the proposed methodology

The approach described in the previous sections is summarised in Figure 5.5. In that case, we don't consider using data from the last reporting period since it produces a higher error.

In summary, when data is not reported:

- If noise contours are provided, cross them with the population grid and calculate the population exposed.
- If contours are not provided, delineate the contour based on the reference contour (see section5.2.3), and then cross with the population grid.

Figure 5.5: Overview of the process for estimating the population exposed to noise from major airports when data is not available.


Estimation of % of **population exposed,** and **population exposed per noise band**
DF4_8 t$_n$
Yes
Has the major airport x data for DF4_8 t$_n$?
No
DF4_8 t$_n$
Select the noise contour model to be replicated
Sum
Has the major airport x noise contours?
No
Adapt the noise contour model to the major airport X
Yes
Cross with the population grid to estimate people exposed
DF4_8 t$_n$
Σ **total** population exposed ±confidence interval
% **exposed per noise band** ±confidence interval
t$_n$ current reporting period
Available data for t$_n$
Data not available for t$_n$ (estimated)

6 Conclusions

This report provides some improvements on gap-filling that could be implemented in the next reporting of END data:

- The selection of the best fit of regression has been systematised and can be replicated with the R script developed in this task. It also provides the results with the corresponding confidence interval by integrating the propagation of errors as a consequence of several steps involved in the final calculation. This is relevant since this process may be very tedious without a proper design of the workflow. It also facilitates completing the workflow in the same environment without the need to change between applications.
- The error of applying European averages has been calculated, being able to assess its accuracy.
- The proposed method for major airports improves the estimate of the people exposed at the cost of more intensive processing.

7 Abbreviations

EEA European Environment Agency

END Environmental Noise Directive

ENDRM Electronic Noise Data Reporting Mechanism

ETC/ATNI European Topic Centre on Air Pollution, Transport, Noise and Industrial Pollution

EU European Union

L_{den} Day-evening-night noise level

L_{night} Night noise level

NAP Noise Action Plans

NOISE Noise Observation and Information Service for Europe

8 References

EU, 2002, Directive 2002/49/EC of the European Parliament and of the Council of 25 June 2002 relating to the assessment and management of environmental noise (OJ L 189, 18.7.2002, p. 12-25).

Fons, J., et al., 2015, Methodological proposal to interpolate a complete coverage on noise exposure at EU level. ETC/ACM Working paper.

Fons, J., et al., 2017, END gap filling data 2012: outcomes' evaluation. Comparison between gap filled data results in 2016 and 2017. ETC/ATM Working paper.

Jones, N., 2013. Forecasting ENDRM DF4_8 data to 2020, 2030 and 2050. Extrium P058.

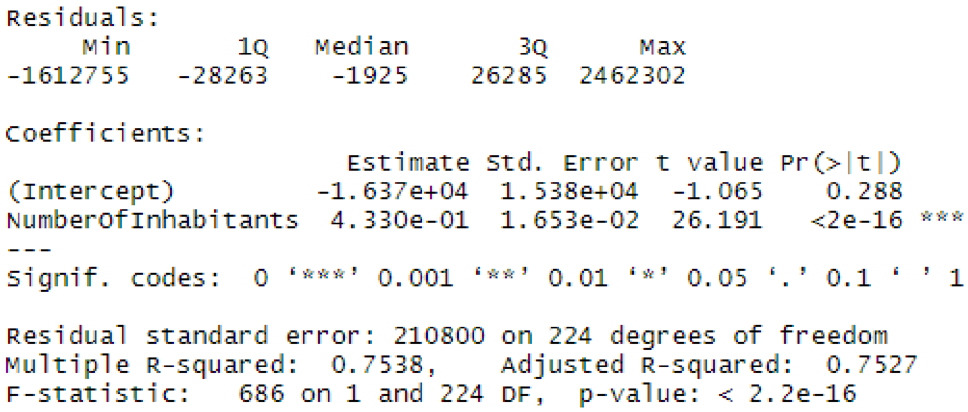
Ramos, M.R., 2019, *Noise indicators under the Environmental Noise Directive. Methodology for estimating missing data*. ETC/ATNI Report 2019/1 (https://www.eionet.europa.eu/etcs/etc-atni/products/etc-atni-reports/noise-indicators-under-the-environmental-noise-directive-methodology-for-estimating-missing-data) accessed 01 July 2021.

Snee, R., 1977, 'Validation of regression models: Methods and Examples', *Technometrics* 19(4), pp. 415-428.

Annex 1
Summary of regression statsitics (agglomeration road)

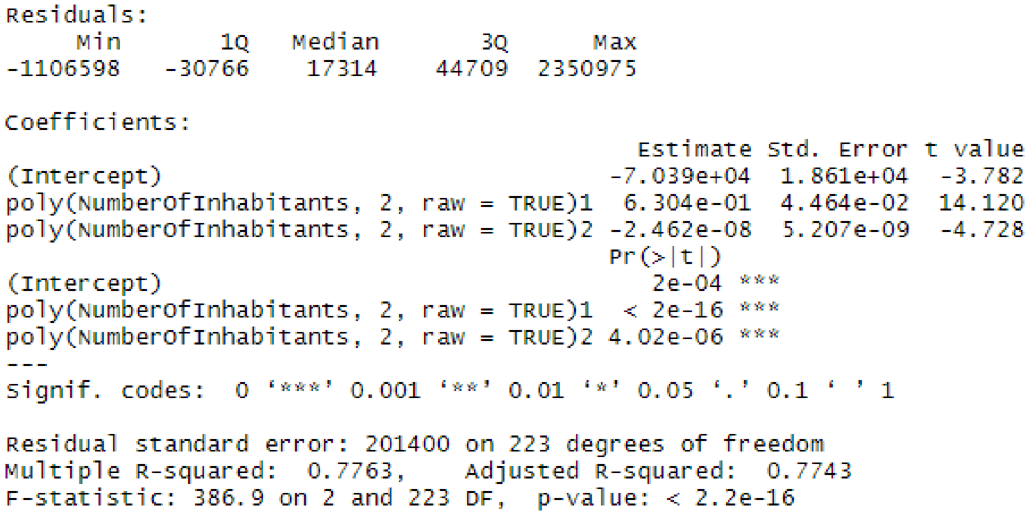
Summary of statistics of the regression between people exposed to noise from roads inside agglomerations (L_{den} >= 55 dB) and Number of inhabitants (see section 1.1.2.4 Validation).

Model 1. Lineal



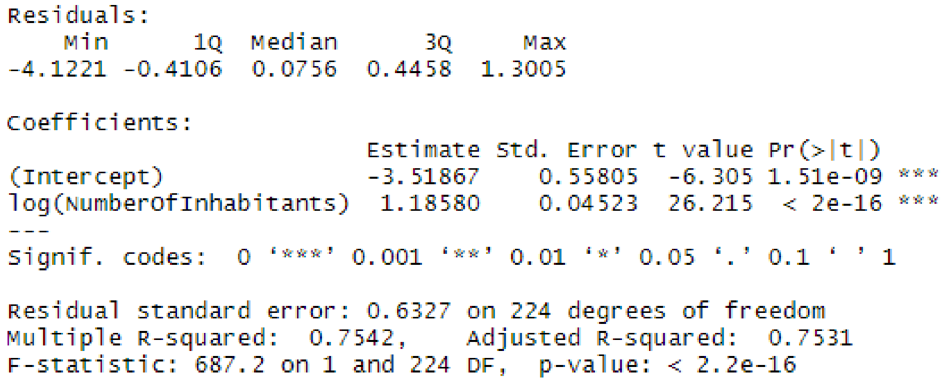
```
Residuals:
     Min       1Q   Median       3Q      Max 
-1612755   -28263    -1925    26285  2462302 

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)    
(Intercept)         -1.637e+04  1.538e+04  -1.065    0.288    
NumberOfInhabitants  4.330e-01  1.653e-02  26.191   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 210800 on 224 degrees of freedom
Multiple R-squared:  0.7538,    Adjusted R-squared:  0.7527 
F-statistic:   686 on 1 and 224 DF,  p-value: < 2.2e-16
```

Model 2. Polynomial (2)



```
Residuals:
     Min       1Q   Median       3Q      Max 
-1106598   -30766    17314    44709  2350975 

Coefficients:
                                          Estimate Std. Error t value
(Intercept)                             -7.039e+04  1.861e+04  -3.782
poly(NumberOfInhabitants, 2, raw = TRUE)1  6.304e-01  4.464e-02  14.120
poly(NumberOfInhabitants, 2, raw = TRUE)2 -2.462e-08  5.207e-09  -4.728
                                        Pr(>|t|)    
(Intercept)                                2e-04 ***
poly(NumberOfInhabitants, 2, raw = TRUE)1  < 2e-16 ***
poly(NumberOfInhabitants, 2, raw = TRUE)2 4.02e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 201400 on 223 degrees of freedom
Multiple R-squared:  0.7763,    Adjusted R-squared:  0.7743 
F-statistic: 386.9 on 2 and 223 DF,  p-value: < 2.2e-16
```

Model 3. Log-log



```
Residuals:
    Min      1Q  Median      3Q     Max 
-4.1221 -0.4106  0.0756  0.4458  1.3005 

Coefficients:
                         Estimate Std. Error t value Pr(>|t|)    
(Intercept)              -3.51867    0.55805  -6.305 1.51e-09 ***
log(NumberOfInhabitants)  1.18580    0.04523  26.215  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6327 on 224 degrees of freedom
Multiple R-squared:  0.7542,    Adjusted R-squared:  0.7531 
F-statistic: 687.2 on 1 and 224 DF,  p-value: < 2.2e-16
```

Annex 2
R script to estimate missing data of people exposed to roads inside agglomerations

R Script (Rnotebook) that process interactively the following steps:

- Import of the data to be processed
- Calculate and exclude outliers
- Builds the total people exposed in Europe
- Select the data reported
- Gap fill with data reported in previous period if available
- Regression for the remaining data
- Subset reported data in test and validation
- Define the models (interactive)
- Test the models
- Select the model (interactive)
- Apply the regression
- Calculate the total with confidence interval

~~~
---
title: "Gap filling of people exposed to Agglomerations roads"
author: "Jaume Fons-Esteve"
date: 04/03/21
output:
  html_notebook: default
  word_document: default
  html_document:
    df_print: paged
    toc: true
    toc_depth: 4
    toc_float: true
editor_options:
  markdown:
    wrap: 72
---

This is an [R Markdown](http://rmarkdown.rstudio.com) Notebook. When you execute code within the notebook,
the results appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and
pressing *Ctrl+Shift+Enter*.

```{r setup, echo=FALSE}
library(knitr)
knitr::opts_chunk$set(echo = FALSE)
```
~~~

~~~
# Overview

This notebook process data to estimate missing values of people exposed to road
noise inside agglomerations. The methodology follows the one described in the report [Methodological
documentation of the gap filling exercise
(2020)](https://uab.sharepoint.com/:w:/s/interfase/EaigFS-
5RqNIqM0HAlCnjH4BFFsL2QQk09j8SKhLFlx5Aw?e=gb2Tb0)

Summary of the process
* year1 clean data: t1.cln
* year2 clean data (current year): t2.cln
* Subset A -> data available: t2A.Ld, t2A.Ln
* Subset B -> missing data, but available at t-1: t2B.Ld
* Subset C -> estimate from regression: t2C.Ld
* At the end merge A+B+C

```{r load }
#Load needed libraries
pacman::p_load(here)
source(here::here("Script", "load_libraries.R"))
```


# 1.Import and prepare the data

Data is imported from Excel. Creates t1 = data from time 1; t2 = data
for time 2 INPUT YEAR

```{r import, echo=FALSE}

#INPUT YEAR_T2 HERE (nsrc)
year_t2 = 17
nsrc = 'road'
year_t1 = ifelse(year_t2 - 5 <10, paste('0', year_t2-5, sep=''), year_t2 - 5 )
nsrc_sheet2 = paste("df4_8_agg_",nsrc,"_",year_t2,sep = "")
nsrc_sheet1 = paste("df4_8_agg_",nsrc,"_",year_t1,sep = "")

#Import data into t1
t1 <- read_excel(choose.files(default = "", caption = "Select file for t1",
        multi = FALSE), sheet = nsrc_sheet1 )

#Import data into t2
t2 <- read_excel(choose.files(default = "", caption = "Select file for t2",
        multi = FALSE), sheet = nsrc_sheet2 )

```


-  Rename noise bands
-  Add a column with the name of the source (Agg_road)
-  Calculate Lden\>=55 (Exclude -2 -1 -9999)
-  Calculate Lnight\>=50 (Exclude -2 -1 -9999)
-  LdRprt = -2, no data; -1 not applicable; 1 reported
-  Ld_part Partially reported (some noise bands -2)
-  Ld_More100Pcnt = TRUE, FALSE LdSum \> NrOfInhabitants
-  Creates new tables t1.cln, t2.cln

```{r clean, echo=FALSE}
#Creates new tables: t1.cln, t2.cln
~~~

```
# t1 -> t1.cln  cln=clean-----------------------------------------------------
# New variables: LdSum, LdRport (reported), Ld_More100Pcnt (LdSum > Inhabitants)
# New variables: LnSum, LnRport (reported), Ln_More100Pcnt (LnSum > Inhabitants)

t1.cln <- t1 %>%
  rename(Ld55 = nLden5559, Ld60 = nLden6064, Ld65 = nLden6569,
         Ld70 = nLden7074, Ld75 = nLden75, Ln50 = nLnight5054,
         Ln55 = nLnight5559, Ln60 = nLnight6064, Ln65 = nLnight6569,
         Ln70 = nLnight70) %>%
  mutate(Src = nsrc)  %>%     #Add a column with the name of the source
  mutate(LdSum = ifelse(Ld55 > -1, Ld55, NA)) %>%
  mutate(LdSum = ifelse(is.na(Ld55), NA, ifelse(Ld60 > -1, LdSum + Ld60, LdSum))) %>% #To exclude nb == -1
  mutate(LdSum = ifelse(is.na(Ld55), NA, ifelse(Ld65 > -1, LdSum + Ld65, LdSum))) %>%
  mutate(LdSum = ifelse(is.na(Ld55), NA, ifelse(Ld70 > -1, LdSum + Ld70, LdSum))) %>%
  mutate(LdSum = ifelse(is.na(Ld55), NA, ifelse(Ld75 > -1, LdSum + Ld75, LdSum))) %>%
  mutate(LnSum = ifelse(Ln50 > -1, Ln50, NA)) %>%
  mutate(LnSum = ifelse(is.na(Ln50), NA, ifelse(Ln55 > -1, LnSum + Ln55, LnSum))) %>%
  mutate(LnSum = ifelse(is.na(Ln50), NA, ifelse(Ln60 > -1, LnSum + Ln60, LnSum))) %>%
  mutate(LnSum = ifelse(is.na(Ln50), NA, ifelse(Ln65 > -1, LnSum + Ln65, LnSum))) %>%
  mutate(LnSum = ifelse(is.na(Ln50), NA, ifelse(Ln70 > -1, LnSum + Ln70, LnSum))) %>%
  mutate(LdRprt = case_when(Ld55 < -1 ~ -2,     #-2 & -9999 assigned  -2
                            Ld55 == -1 ~ -1,    #-1 assigned -1
                            Ld55 > -1 ~ 1)) %>%
  mutate(LnRprt = case_when(Ln50 < -1 ~ -2,     #-2 & -9999 assigned  -2
                            Ln50 == -1 ~ -1,    #-1 assigned -1
                            Ln50 > -1 ~ 1)) %>%
  mutate(Ld_More100Pcnt = if_else(LdSum > NumberOfInhabitants ,TRUE, FALSE)) %>%
  mutate(Ln_More100Pcnt = if_else(LnSum > NumberOfInhabitants ,TRUE, FALSE))

#Identify partial gap filling t1 Lden
clmn_55 <- fmatch("Ld55",names(t1.cln)) #first column for noise bands Lden. Use column index instead column
name to facilitate loop below.
clmn_75 <- clmn_55 + 4
n <- 0 #Used to identify if we are on the first step on the loop
for (i in clmn_55:clmn_75)  #Loop thorough all the Lden noise bands
{
  nb <- i -clmn_55 +1   #When a value for a noise band is missing Ld_part takes the index nb (1 to 5)
  if (n == 0) {         #Lden55
    t1.cln <- t1.cln %>%
    mutate(Ld_part = ifelse(LdRprt == 1 &  .[[i]] == -2, nb, '0'))
    n <- 1
} else {
    t1.cln <- t1.cln %>%
    mutate(Ld_part = ifelse(LdRprt == 1 & .[[i]] == -2, paste(Ld_part, nb), paste(Ld_part,'0')))
}

}

#Identify partial gap filling t1 Ln
clmn_50 <- fmatch("Ln50",names(t1.cln)) #first column for noise bands Lden. Use column index instead column
name to facilitate loop below.
clmn_70 <- clmn_50 + 4
n <- 0 #Used to identify if we are on the first step on the loop
for (i in clmn_50:clmn_70)  #Loop thorough all the Lden noise bands
{
  nb <- i -clmn_50 +1   #When a value for a noise band is missing Ln_part takes the index nb (1 to 5)
  if (n == 0) {         #Lden55
t1.cln <- t1.cln %>%
  mutate(Ln_part = ifelse(LnRprt == 1 &  .[[i]] == -2, nb, '0'))
n <- 1
} else {
t1.cln <- t1.cln %>%
  mutate(Ln_part = ifelse(LnRprt == 1 & .[[i]] == -2, paste(Ln_part, nb), paste(Ln_part,'0')))
}

}
# t1  Reorder columns, exclude extra info
t1.cln <- t1.cln[c("Src","Ctry","Ctry2", "ReferenceYear", "RLID",  "AggloNameEn",
                "EU28", "NumberOfInhabitants", "Ld55", "Ld60", "Ld65", "Ld70",
                "Ld75", "LdSum", "LdRprt", "Ld_part", "Ld_More100Pcnt", "Ln50", "Ln55",                        "Ln60", "Ln65",
"Ln70", "LnSum", "LnRprt", "Ln_part", "Ln_More100Pcnt")]
# t2 -------------------------------------------------------
t2.cln <- t2 %>%
  rename(Ld55 = nLden5559, Ld60 = nLden6064, Ld65 = nLden6569, Ld70 = nLden7074,
       Ld75 = nLden75, Ln50 = nLnight5054, Ln55 = nLnight5559,
       Ln60 = nLnight6064, Ln65 = nLnight6569, Ln70 = nLnight70) %>%
  mutate(Src = nsrc)  %>%   #create a column with the name of the source
  mutate(LdSum = ifelse(Ld55 > -1, Ld55, NA)) %>%
  mutate(LdSum = ifelse(is.na(Ld55), NA, ifelse(Ld60 > -1, LdSum + Ld60, LdSum))) %>% #To exclude nb == -1
  mutate(LdSum = ifelse(is.na(Ld55), NA, ifelse(Ld65 > -1, LdSum + Ld65, LdSum))) %>%
  mutate(LdSum = ifelse(is.na(Ld55), NA, ifelse(Ld70 > -1, LdSum + Ld70, LdSum))) %>%
  mutate(LdSum = ifelse(is.na(Ld55), NA, ifelse(Ld75 > -1, LdSum + Ld75, LdSum))) %>%
  mutate(LnSum = ifelse(Ln50 > -1, Ln50, NA)) %>%
  mutate(LnSum = ifelse(is.na(Ln50), NA, ifelse(Ln55 > -1, LnSum + Ln55, LnSum))) %>%
  mutate(LnSum = ifelse(is.na(Ln50), NA, ifelse(Ln60 > -1, LnSum + Ln60, LnSum))) %>%
  mutate(LnSum = ifelse(is.na(Ln50), NA, ifelse(Ln65 > -1, LnSum + Ln65, LnSum))) %>%
  mutate(LnSum = ifelse(is.na(Ln50), NA, ifelse(Ln70 > -1, LnSum + Ln70, LnSum))) %>%
  mutate(LdRprt = case_when(Ld55 < -1 ~ -2,      #-2 & -9999 assigned  -2
                    Ld55 == -1 ~ -1,     #-1 assigned -1
                    Ld55 > -1 ~ 1)) %>%
  mutate(LnRprt = case_when(Ln50 < -1 ~ -2,  #-2 & -9999 assigned  -2
                    Ln50 == -1 ~ -1,     #-1 assigned -1
                    Ln50 > -1 ~ 1)) %>%
  mutate(Ld_More100Pcnt = if_else(LdSum > NumberOfInhabitants ,TRUE, FALSE)) %>%
  mutate(Ln_More100Pcnt = if_else(LnSum > NumberOfInhabitants ,TRUE, FALSE))

#Identify partial gap filling t2 Lden
clmn_55 <- fmatch("Ld55",names(t2.cln)) #first column for noise bands Lden. Use column index instead column
name to facilitate loop below.
clmn_75 <- clmn_55 + 4
n <- 0 #Used to identify if we are on the first step on the loop
for (i in clmn_55:clmn_75)  #Loop thorugh all the Lden noise bands
{
  nb <- i -clmn_55 +1   #When a value for a noise band is missing Ld_part takes the index nb (1 to 5)
  if (n == 0) {         #Lden55
t2.cln <- t2.cln %>%
  mutate(Ld_part = ifelse(LdRprt == 1 &  .[[i]] == -2, nb, '0'))
n <- 1
} else {
t2.cln <- t2.cln %>%
  mutate(Ld_part = ifelse(LdRprt == 1 & .[[i]] == -2, paste(Ld_part, nb), paste(Ld_part,'0')))
}

}
```

```
#Identify partial gap filling t1 Ln
clmn_50 <- fmatch("Ln50",names(t2.cln)) #first column for noise bands Lden. Use column index instead column
name to facilitate loop below.
clmn_70 <- clmn_50 + 4
n <- 0 #Used to identify if we are on the first step on the loop
for (i in clmn_50:clmn_70)  #Loop thorough all the Lden noise bands
{
  nb <- i -clmn_50 +1   #When a value for a noise band is missing Ln_part takes the index nb (1 to 5)
  if (n == 0) {          #Lden55
t2.cln <- t2.cln %>%
  mutate(Ln_part = ifelse(LnRprt == 1 &  .[[i]] == -2, nb, '0'))
n <- 1
} else {
t2.cln <- t2.cln %>%
  mutate(Ln_part = ifelse(LnRprt == 1 & .[[i]] == -2, paste(Ln_part, nb), paste(Ln_part,'0')))
}

}

# t2   Reorder columns, exclude extra info
t2.cln <- t2.cln[c("Src","Ctry","Ctry2", "ReferenceYear", "RLID",  "AggloNameEn",
                "EU28", "NumberOfInhabitants", "Ld55", "Ld60", "Ld65", "Ld70",
                "Ld75",  "LdSum",  "LdRprt",    "Ld_part","Ld_More100Pcnt",  "Ln50",  "Ln55",  "Ln60",  "Ln65",  "Ln70",
"LnSum", "LnRprt", "Ln_part", "Ln_More100Pcnt")]
rm(clmn_50,clmn_55, clmn_70, clmn_75, i, n, nb)
```

```{r pviot_reported, echo=FALSE}
t1.Ld.pt <- PivotTable$new()  #Pivot table with counts of missing data
t1.Ld.pt$addData(t1.cln)
t1.Ld.pt$addColumnDataGroups("LdRprt")
t1.Ld.pt$addRowDataGroups("Ctry")
t1.Ld.pt$defineCalculation(calculationName="Total", summariseExpression="n()")
t1.Ld.pt$evaluatePivot()

t1.Ln.pt <- PivotTable$new()  #Pivot table with counts of missing data
t1.Ln.pt$addData(t1.cln)
t1.Ln.pt$addColumnDataGroups("LnRprt")
t1.Ln.pt$addRowDataGroups("Ctry")
t1.Ln.pt$defineCalculation(calculationName="Total", summariseExpression="n()")
t1.Ln.pt$evaluatePivot()


t2.Ld.pt <- PivotTable$new()  #Pivot table with counts of missing data
t2.Ld.pt$addData(t2.cln)
t2.Ld.pt$addColumnDataGroups("LdRprt")
t2.Ld.pt$addRowDataGroups("Ctry")
t2.Ld.pt$defineCalculation(calculationName="Total_Ld", summariseExpression="n()")
t2.Ld.pt$evaluatePivot()

t2.Ln.pt <- PivotTable$new()  #Pivot table with counts of missing data
t2.Ln.pt$addData(t2.cln)
t2.Ln.pt$addColumnDataGroups("LnRprt")
t2.Ln.pt$addRowDataGroups("Ctry")
t2.Ln.pt$defineCalculation(calculationName="Total_Ln", summariseExpression="n()")
```

~~~
t2.Ln.pt$evaluatePivot()
```

t1 = previous reporting period, t2 = current reporting period

```{r pivot_reported_tbl, echo=TRUE}
t1.Ld.pt
t1.Ln.pt
t2.Ld.pt
t2.Ln.pt
rm(t1.Ld.pt, t1.Ln.pt,t2.Ld.pt,t2.Ln.pt)
```

## Output: agglomerations where Ldn, Ln \> NrInhabitants. NrOfInhabitants < 100.000

```{r MoreThan100, echo=TRUE}
t1.cln %>% filter(Ld_More100Pcnt == TRUE) %>%
  select(Ctry, AggloNameEn, NumberOfInhabitants,LdSum) %>%
  arrange(Ctry, AggloNameEn)

t1.cln %>% filter(Ln_More100Pcnt == TRUE) %>%
  select(Ctry, AggloNameEn, NumberOfInhabitants,LnSum) %>%
  arrange(Ctry, AggloNameEn)

t2.cln %>% filter(Ld_More100Pcnt == TRUE) %>%
  select(Ctry, AggloNameEn, NumberOfInhabitants,LdSum) %>%
  arrange(Ctry, AggloNameEn)

t2.cln %>% filter(Ln_More100Pcnt == TRUE) %>%
  select(Ctry, AggloNameEn, NumberOfInhabitants,LnSum) %>%
  arrange(Ctry, AggloNameEn)

t2.cln %>% filter(NumberOfInhabitants < 100000) %>%
  select(Ctry, Ctry2, AggloNameEn, ReferenceYear, NumberOfInhabitants) %>%
  arrange(NumberOfInhabitants)

```
# 2. Gap filling
There are three steps
* Subset A. Data available for t2 (current phase)
* Subset B. Data not available for t2, but available from t1
* Subset C. None of the previous cases apply. Regression

## 2.1.A.Subset of available data for t2

Output: t2A.Ld, t2A.Ln\
t2 = current phase\
A = data reported\
New columns Ld_GapFilled & Ld_Change

Agglomerations where people exposed \> NrInhabitants:

```{r subsetA}

t2A.Ld <- t2.cln %>%
      filter(LdRprt > -2) %>%     #data reported for t2 (including -1)
      filter(Ld_part == '0 0 0 0 0') %>% #NO partial gap filling
~~~

~~~~
    mutate(Ld_GapFilled = paste('No gap filling -data from 20',year_t2,sep="")) %>%   #create column DataSrce
    mutate(Ld_Change = 'No change') %>%
    select(-18:-26)

t2A.Ln <- t2.cln %>%
    filter(LnRprt > -2) %>%     #data reported for t2
    filter(Ln_part == '0 0 0 0 0') %>% #NO partial gap filling
    mutate(Ln_GapFilled = paste('No gap filling -data from 20',year_t2, sep="")) %>%   #create column DataSrce
    mutate(Ln_Change = 'No change') %>%
    select(-9:-17)   #exclude Ld

#Agglomerations where People exposed > NrInhabitants
t2A.Ld %>% filter(Ld_More100Pcnt == TRUE) %>%
 select(Ctry, AggloNameEn, NumberOfInhabitants,LdSum) %>%
 arrange(Ctry, AggloNameEn)

t2A.Ln %>% filter(Ln_More100Pcnt == TRUE) %>%
 select(Ctry, AggloNameEn, NumberOfInhabitants,LnSum) %>%
 arrange(Ctry, AggloNameEn)
```

## 2.2.B.Agg without data for t2 and data available for t1: use data from t1

Output: t2B.Ld, t2B.\
Ln t2 = current phase \
B = data from t1 (previous reporting cycle) \
New columns Ld_GapFilled & Ld_Change

Tables: Agglomerations where people exposed \> NrInhabitants:

```{r subsetB, echo=TRUE}

#Lden------------------------
tmp1.Ld <- t1.cln %>%
      filter(LdRprt > -2) %>%     #Select from t1 reported data, also -1
      select(-15, -16,-18:-26)       #Exclude columns from Ln, LnRprt
tmp2.Ld <- t2.cln %>%
      filter(LdRprt == -2) %>%  #select from t2 missing data (-2 includes -9999)
      select("Ctry", "Ctry2","RLID", "LdRprt", "Ld_part")
t2B.Ld <- inner_join(tmp1.Ld, tmp2.Ld, by = c("Ctry", "Ctry2", "RLID")) %>%     #only agglomerations common in both tables
      mutate(Ld_GapFilled = paste('Data from 20',year_t1, sep="")) %>%   #create column DataSrce
      mutate(Ld_Change = 'Change')

t2B.Ld <- t2B.Ld[, c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,16,17,15,18,19)] #Reorder columns

#Lnight--------------------------
tmp1.Ln <- t1.cln %>%              #Select from t1 reported data, also -1
      filter(LnRprt >-2) %>%
      select(-9:-17, -24,-25)     #Exclude columns from Ld, LnRprt
tmp2.Ln <- t2.cln %>%
      filter(LnRprt == -2) %>%  #select from t2 missing data
      select("Ctry", "Ctry2","RLID", "LnRprt", "Ln_part")
t2B.Ln <- inner_join(tmp1.Ln, tmp2.Ln, by = c("Ctry", "Ctry2", "RLID")) %>%
      mutate(Ln_GapFilled = paste('Data from 20',year_t1, sep="")) %>%   #create column DataSrce
      mutate(Ln_Change = 'Change')
~~~~

~~~
#Agglomerations where People exposed > NrInhabitants
t2B.Ld %>% filter(Ld_More100Pcnt == TRUE) %>%
 select(Ctry, AggloNameEn, NumberOfInhabitants,LdSum) %>%
 arrange(Ctry, AggloNameEn)

t2B.Ln %>% filter(Ln_More100Pcnt == TRUE) %>%
 select(Ctry, AggloNameEn, NumberOfInhabitants,LnSum) %>%
 arrange(Ctry, AggloNameEn)

rm(tmp1.Ld, tmp2.Ld, tmp1.Ln, tmp2.Ln)
```


## 2.3 C.Regression

 Data not available (t1 neither t2)
 Regression  population exposed = f(NumberOfInhabitants)

### 230 Create subset C. Data to be gapfilled with regression
```{r subsetCLdLn}
#Ld. Select agglomerations to be estimated. Based on excluding agglomerations in A & B---
tmpA <- t2A.Ld %>%
      select('Ctry','Ctry2','RLID')
tmpB <- t2B.Ld %>%
      select('Ctry','Ctry2','RLID')
tmpAB <- rbind(tmpA, tmpB) #List of agglomerations A + B

t2C.Ld <- anti_join(t2.cln, tmpAB, by=c("Ctry", "Ctry2","RLID")) %>%  #select agglomerations not included A & B
         select(-18:-26)

#Ln. Select agglomerations to be estimated. Based on excluding agglomerations in A & B---
tmpA <- t2A.Ln %>%
      select('Ctry','Ctry2','RLID')
tmpB <- t2B.Ln %>%
      select('Ctry','Ctry2','RLID')
tmpAB <- rbind(tmpA, tmpB) #List of agglomerations A + B

t2C.Ln <- anti_join(t2.cln, tmpAB, by=c("Ctry", "Ctry2","RLID")) %>%  #select agglomerations not included A & B
         select(-9:-17)

rm(tmpA, tmpAB, tmpB)
```


### 231 Outliers from % change population exposed

#### a. Join t1.cln2 & t2.cln2 (cln = clean) needed to identify outliers

New table created: t1t2

Output table: rows t1, columns t2. Parameters: Reported (-2, -1, 1),
MoreThan1000Pcnt (TRUE, FALSE, NA)

```{r join_t1_t2}
# join t1 & t2 (only agglomerations in t1 & t2)
t1t2 <- inner_join(t1.cln, t2.cln, by = c("Ctry", "Ctry2", "RLID"), suffix = c(".t1", ".t2"))

# create difference and % of difference [100*(t2-t1)/t1]
t1t2 <- t1t2 %>%
~~~

```
 mutate(ld.t2.t1 = LdSum.t2 - LdSum.t1) %>%
 mutate(ln.t2.t1 = LnSum.t2 - LnSum.t1) %>%
 mutate(ld.pcnt = 100 * ld.t2.t1 / LdSum.t1) %>%  #percentage of difference
 mutate(ln.pcnt = 100 * ln.t2.t1 / LnSum.t1)
```

```{r join.pivot}
#Pivot tables for counting joint data t1 t2
t1t2.Ld.pt <- PivotTable$new()  #Pivot table with counts of missing data
t1t2.Ld.pt$addData(t1t2)
t1t2.Ld.pt$addColumnDataGroups("LdRprt.t2")
t1t2.Ld.pt$addColumnDataGroups("Ld_More100Pcnt.t2")
t1t2.Ld.pt$addRowDataGroups("LdRprt.t1")
t1t2.Ld.pt$addRowDataGroups("Ld_More100Pcnt.t1")

t1t2.Ld.pt$defineCalculation(calculationName="Total", summariseExpression="n()")
t1t2.Ld.pt$evaluatePivot()

t1t2.Ld.pt
#Scheme for the output:
#           LdRprt.t2  (-2= not reported, -1= not applicable, 1= reported)
#           Ld_More100Pcnt.t2 (NA = no data, TRUE = > 100% , FALSE = below 100%
#
#LdRprt.t1 Ld_More100Pcnt.t1
#
```

#### b. Outliers of change t2-t1 (%)

Create columns Outl_Ld, Outl_Ln (TRUE, FALSE)

```{r outl}

# Statistics for Lden % of change (interquartile range to identify outliers)--------------
tmpt1t2.Ld <- t1t2 %>%        #exclude SumLden > NInhabitants
 filter(LdRprt.t1 == 1)  %>%        #only reported
 filter(LdRprt.t2 == 1)  %>%        #only reported
 filter(Ld_More100Pcnt.t1 == FALSE & Ld_More100Pcnt.t2 == FALSE) %>%
 select(ld.pcnt)
Ld.summary <- summary(tmpt1t2.Ld$ld.pcnt)
iqrLd <- Ld.summary[[5]] - Ld.summary[[2]] # Estimate interquartile range (3rd - 1st)
lower_bound_Ld <- Ld.summary[[2]] - (1.5 * iqrLd) # Identify bounds for outliers
upper_bound_Ld <- Ld.summary[[5]] + (1.5 * iqrLd)

# create column outlier in t1t2
t1t2 <- t1t2 %>%
 mutate(Outl_Ld = if_else(ld.pcnt > upper_bound_Ld | ld.pcnt <lower_bound_Ld, 'yes', 'no'))

# Statistics for Lnight-------------------------------------------------
tmpt1t2.Ln <- t1t2 %>%
 filter(LnRprt.t1 == 1)  %>%        #only reported
 filter(LnRprt.t2 == 1)  %>%        #only reported
 filter(Ln_More100Pcnt.t1 == FALSE & Ln_More100Pcnt.t2 == FALSE) %>%
 select(ln.pcnt)
Ln.summary <- summary(tmpt1t2.Ln$ln.pcnt)
iqrLn <- Ln.summary[[5]] - Ln.summary[[2]] # Estimate interquartile range (3rd - 1st)
lower_bound_Ln <- Ln.summary[[2]] - (1.5 * iqrLn)  # Identify bounds for outliers
```

~~~
upper_bound_Ln <- Ln.summary[[5]] + (1.5 * iqrLn)

# create column outlier
t1t2 <- t1t2 %>%
  mutate(Outl_Ln = if_else(ln.pcnt > upper_bound_Ln | ln.pcnt < lower_bound_Ln, 'yes', 'no'))

# add the columns ld.pcnt, ln.pcnt, Outl_Ld, Outl_Ln to t2.cln--------------
tmpt1t2 <- t1t2 %>% select("Ctry", "Ctry2", "RLID", "ld.pcnt", "ln.pcnt", "Outl_Ld", "Outl_Ln") #select only relevant columns
t2.cln  <- left_join(t2.cln, tmpt1t2, by= c("Ctry", "Ctry2","RLID") )    #add columns to t2.cln

#rename Outl_Ld Outl_Ln those agglomerations with no data in t1
t2.cln <- t2.cln %>%
    mutate(Outl_Ld=replace(Outl_Ld, is.na(Outl_Ld) & Ld55 > -1, 'No_data_t1')) %>%
    mutate(Outl_Ln=replace(Outl_Ln, is.na(Outl_Ln) & Ln50 > -1, 'No_data_t1'))

t2A.Ld  <- left_join(t2A.Ld, tmpt1t2, by= c("Ctry", "Ctry2","RLID") ) %>%    #add columns to t2A.Ld
    mutate(Outl_Ld=replace(Outl_Ld, is.na(Outl_Ld) & Ld55 > -1, 'No_data_t1')) %>%
    select(-21, -23)

t2A.Ln  <- left_join(t2A.Ln, tmpt1t2, by= c("Ctry", "Ctry2","RLID") )  %>%     #add columns to t2A.Ln
    mutate(Outl_Ln=replace(Outl_Ln, is.na(Outl_Ln) & Ln50 > -1, 'No_data_t1')) %>%
    select(-20, -22)
```

######  Summary of statistics (% of change t1 -> t2)

```{r outl_output, echo=TRUE}
count(tmpt1t2.Ld)
count(t1t2 %>%
        filter(Outl_Ld == TRUE))
Ld.summary
iqrLd
lower_bound_Ld
upper_bound_Ld
count(tmpt1t2.Ln)
count(t1t2 %>%
        filter(Outl_Ln == TRUE))
Ln.summary
iqrLn
lower_bound_Ln
upper_bound_Ln

rm(tmpt1t2, tmpt1t2.Ld, tmpt1t2.Ln)
```

######  Histogram % of change

```{r histogram, evalu=TRUE}
tmp2.Ld <- t2.cln %>%
       filter(Ld_More100Pcnt == FALSE) #exclude Lden > NrInhabitants
#plot histogram with outliers
ggplot(tmp2.Ld, aes(x=ld.pcnt, fill=Outl_Ld)) +
  geom_histogram(binwidth=10, alpha=.7, position="identity") +
  scale_x_continuous(labels = comma_format(big.mark = ".",
              decimal.mark = ",")) +  #change scientific notation
  xlab("Lden (difference t2-t1 (%))") +      #x label
~~~

```
  labs(fill = "Outliers")            #title of legend

tmp2.Ln <- t2.cln %>%
         filter(Ln_More100Pcnt == FALSE) #exclude Ln > NrInhabitants
ggplot(tmp2.Ln, aes(x=ln.pcnt, fill=Outl_Ln)) +
  geom_histogram(binwidth=10, alpha=.9, position="identity") +
  scale_x_continuous(labels = comma_format(big.mark = ".",
              decimal.mark = ",")) +  #change scientific notation
  xlab("Lnight (difference t2-t1 (%))") +       #x label
  labs(fill = "Outliers")            #title of legend

rm(tmp2.Ld, tmp2.Ln)
```
~~~
### 232 Outliers t2 SumLden >= 55 dB, SumLn >= 50 dB

t2 exclude >100% exposed
t2 identify outliers LdSum LnSum
```{r OutlRgr}

#Lden Statistics
tmp2.Ld <- t2A.Ld %>%         #exclude SumLden > NInhabitants
  filter(Ld_More100Pcnt == FALSE) %>%
  filter(Outl_Ld == 'no' | Outl_Ld == 'No_data_t1') %>%
  select(LdSum)
Ld.summary <- summary(tmp2.Ld$LdSum)
iqrLd <- Ld.summary[[5]] - Ld.summary[[2]] # Estimate interquartile range (3rd - 1st)
lower_bound_Ld <- Ld.summary[[2]] - (1.5 * iqrLd) # Identify bounds for outliers
upper_bound_Ld <- Ld.summary[[5]] + (1.5 * iqrLd)

#Lden create column outlier
t2A.Ld <- t2A.Ld %>%
  mutate(Outl_LdSum = if_else(LdSum > upper_bound_Ld | LdSum <lower_bound_Ld, TRUE, FALSE))

#Lnight Statistics
tmp2.Ln <- t2A.Ln %>%
  filter(Ln_More100Pcnt == FALSE) %>%
  filter(Outl_Ln == 'no' | Outl_Ln == 'No_data_t1') %>%
  select(LnSum)
Ln.summary <- summary(tmp2.Ln$LnSum)
iqrLn <- Ln.summary[[5]] - Ln.summary[[2]] # Estimate interquartile range (3rd - 1st)
lower_bound_Ln <- Ln.summary[[2]] - (1.5 * iqrLn)  # Identify bounds for outliers
upper_bound_Ln <- Ln.summary[[5]] + (1.5 * iqrLn)

#Lnight create column outlier
t2A.Ln <- t2A.Ln %>%
  mutate(Outl_LnSum = if_else(LnSum > upper_bound_Ln | LnSum < lower_bound_Ln, TRUE, FALSE))
rm(tmp2.Ld, tmp2.Ln)
```
~~~

#### a.Table with statistics and thresholds for outliers
```{r outl_output_Rgr, echo=TRUE}
Ld.summary
iqrLd
lower_bound_Ld
upper_bound_Ld
Ln.summary
```

~~~
iqrLn
lower_bound_Ln
upper_bound_Ln
```

#### b.Histograms with outliers
```{r histogram_Rgr, evalu=TRUE}
tmp2.Ld <- t2A.Ld %>%
      filter(Ld_More100Pcnt == FALSE) %>%
      filter(Outl_Ld == 'no' | Outl_Ld == 'No_data_t1') %>%
      filter(NumberOfInhabitants > -1)
ggplot(tmp2.Ld, aes(x=LdSum, fill=Outl_LdSum)) +
  geom_histogram(binwidth=100000, alpha=.8, position="identity") +
  scale_x_continuous(labels = comma_format(big.mark = ".",
              decimal.mark = ",")) +  #change scientific notation
   xlab("Lden (People exposed Lden >= 55)") +       #x label
  labs(fill = "Outliers")           #title of legend

tmp2.Ln <- t2A.Ln %>%
     filter(Ln_More100Pcnt == FALSE) %>%
     filter(Outl_Ln == 'no' | Outl_Ln == 'No_data_t1') %>%
     filter(NumberOfInhabitants > -1)
ggplot(tmp2.Ln, aes(x=LnSum, fill=Outl_LnSum)) +
  geom_histogram(binwidth=25000, alpha=.8, position="identity") +
  scale_x_continuous(labels = comma_format(big.mark = ".",
              decimal.mark = ",")) +  #change scientific notation
  xlab("Lnight (People exposed Lnight >= 50)") +       #x label
  labs(fill = "Outliers")           #title of legend
rm(tmp2.Ld, tmp2.Ln)
```
#### c. All outliers
```{r Outliers_all}
#plot all outliers
t2A.Ld <- t2A.Ld %>%
  mutate(Outlier_all = ifelse(Outl_Ld != 'yes' & Outl_LdSum == FALSE, FALSE, TRUE) )

t2A.Ln <- t2A.Ln %>%
  mutate(Outlier_all = ifelse(Outl_Ln != 'yes' & Outl_LnSum == FALSE, FALSE, TRUE) )

#plot histogram with outliers
ggplot(t2A.Ld, aes(x=LdSum, fill=Outlier_all)) +
  geom_histogram(binwidth=50000, alpha=.7, position="identity") +
  scale_x_continuous(labels = comma_format(big.mark = ".",
              decimal.mark = ",")) +  #change scientific notation
  xlab("Lden (difference t2-t1 (%))") +       #x label
  labs(fill = "Outliers")           #title of legend

ggplot(t2A.Ld, aes(x=NumberOfInhabitants, y=LdSum, color=Outlier_all)) +
  geom_point() +
  scale_x_continuous(labels = comma_format(big.mark = ".",
              decimal.mark = ",")) +  #change scientific notation
  xlab("Number of inhabitants") +       #x label
  labs(fill = "Outliers")           #title of legend

#plot histogram with outliers
ggplot(t2A.Ln, aes(x=LnSum, fill=Outlier_all)) +
  geom_histogram(binwidth=50000, alpha=.7, position="identity") +
~~~

```
  scale_x_continuous(labels = comma_format(big.mark = ".",
              decimal.mark = ",")) +  #change scientific notation
  xlab("Lnight (difference t2-t1 (%))") +       #x label
  labs(fill = "Outliers")            #title of legend

ggplot(t2A.Ln, aes(x=NumberOfInhabitants, y=LnSum, color=Outlier_all)) +
  geom_point() +
  scale_x_continuous(labels = comma_format(big.mark = ".",
              decimal.mark = ",")) +  #change scientific notation
  xlab("Number of inhabitants") +       #x label
  labs(fill = "Outliers")            #title of legend
```

### 233 Regression Lden

 regression
   iterate for each model
     regression
     statistics of regression
     plot residuals
 plot outcome of different models
 estimate for missing data with choosen model

#### a.Plot Lden ~ Nr of inhabitants
 look at the scatter plot to decide alternative models
```{r plotLdPop}
#Lden Plot with outliers t2
tmp2.Ld <- t2A.Ld %>%
  filter(Ld_More100Pcnt == FALSE &
         NumberOfInhabitants > -1 &
         Outlier_all == FALSE)
  ggplot(data = t2A.Ld, mapping = aes(x = NumberOfInhabitants, y = LdSum)) +
  geom_point() +
  scale_x_continuous(labels = comma_format(big.mark = ".",
                                    decimal.mark = ",")) +
  scale_y_continuous(labels = comma_format(big.mark = ".",
                                    decimal.mark = ","))

#Lden Plot without outliers
tmp2.Ld <- t2A.Ld %>% filter(Ld_More100Pcnt == FALSE &
                        NumberOfInhabitants > -1 &
                        Outlier_all == FALSE)
ggplot(data = tmp2.Ld, mapping = aes(x = NumberOfInhabitants, y = LdSum)) +
  geom_point() +
  scale_x_continuous(labels = comma_format(big.mark = ".",
                                    decimal.mark = ",")) +
  scale_y_continuous(labels = comma_format(big.mark = ".",
                                    decimal.mark = ","))

t2A.Ld %>% filter(Ld_More100Pcnt == FALSE & NumberOfInhabitants > -1) %>%
                     count()
t2A.Ld %>% filter(Ld_More100Pcnt == FALSE &
              NumberOfInhabitants > -1 & Outlier_all == FALSE) %>% count()
rm(tmp2.Ld)
```

~~~
#### b.Transformations

Transform LdenSum to log

```{r log_trans}
#Transform if it is needed
t2A.Ld <- t2A.Ld %>%
        mutate(log.LdSum = log(LdSum))

tmp2A <- t2A.Ld %>% filter(Ld_More100Pcnt == FALSE &
                NumberOfInhabitants > -1 &
                Outlier_all == FALSE)

ggplot(data = tmp2A, mapping = aes(x = NumberOfInhabitants, y = log.LdSum)) +
  geom_point() +
  scale_x_continuous(labels = comma_format(big.mark = ".",
                      decimal.mark = ",")) +
  scale_y_continuous(labels = comma_format(big.mark = ".",
                      decimal.mark = ","))

~~~

transform sqrt
```{r sqrt_trans}
#Transform if it is needed
t2A.Ld <- t2A.Ld %>%
        mutate(sqrt(LdSum))

tmp2A <- t2A.Ld %>% filter(Ld_More100Pcnt == FALSE &
                NumberOfInhabitants > -1 &
                Outlier_all == FALSE)

ggplot(data = t2A.Ld, mapping = aes(x = NumberOfInhabitants, y = sqrt(LdSum))) +
  geom_point() +
  scale_x_continuous(labels = comma_format(big.mark = ".",
                      decimal.mark = ",")) +
  scale_y_continuous(labels = comma_format(big.mark = ".",
                      decimal.mark = ","))

```

Transform loglog
```{r loglog_trans}
#Transform if it is needed
t2A.Ld <- t2A.Ld %>%
        mutate(log(NumberOfInhabitants))

tmp2A <- t2A.Ld %>% filter(Ld_More100Pcnt == FALSE &
                NumberOfInhabitants > -1 &
                Outlier_all == FALSE)

ggplot(data = tmp2A, mapping = aes(x = log(NumberOfInhabitants), y = log(LdSum))) +
  geom_point() +
  scale_x_continuous(labels = comma_format(big.mark = ".",
                      decimal.mark = ",")) +
  scale_y_continuous(labels = comma_format(big.mark = ".",
```

~~~~
                                   decimal.mark = ","))
```

#### c.Random selection of a subset for verification
Output: t2A.Ld.model to run the regression; t2A.Ld.validation to validate

Random selection filtering by range of NumberOfInhabitants

```{r rangeNrInhb}
#Select the same range of NumberOfInhabitants than the missing data

# Statistics & histogram of aggl to be gap filled
t2C.Ld %>%
 select(NumberOfInhabitants) %>%
 filter(NumberOfInhabitants > -1) %>%   #Only if NrOfInhabitants available
 summarise(n(), min(., na.rm = TRUE), max(., na.rm = TRUE), mean(NumberOfInhabitants, na.rm = TRUE) )

t2C.Ld %>%
 select(NumberOfInhabitants) %>%
 filter(NumberOfInhabitants > -1) %>%   #Only if NrOfInhabitants available
 {ggplot(., aes(x=NumberOfInhabitants)) +
 geom_histogram(binwidth=25000, fill="#69b3a2", color="#e9ecef") +
 scale_x_continuous(labels = comma_format(big.mark = ".",
              decimal.mark = ","))}

```

INTRODUCE THE RANGE OF NUMBER OF INHABITANTS
Test Kolmogorov_Smirnov p as high as possible
```{r splitModelvalidation}
#random selection ENTER THE RANGE OF NUMBER OF INHABITANTS
#Select 70% of all aggl for regression, the rest for validation
#GC-->S'ha de marcar un rang de valors relatiu al nombre d'habitans que sigui coherent amb el rang de valors de la taula anterior (tb1_dbf) donant marge per els dos costats
#GC--> El resulat d'aquest procès ens ha de donar una p-value per sobre del 0.7. Repetir fins que aixó sigui així.
t2A.Ld.NrInh <- t2A.Ld %>%
 filter(Ld_More100Pcnt == FALSE) %>%
 filter(Outlier_all == FALSE) %>%
 filter(NumberOfInhabitants >= 60000 & NumberOfInhabitants <= 850000)
#ENTER THE NUMBER OF AGGLOMERATIONS
nAggl <- round(nrow(t2A.Ld.NrInh)*0.30)
nAggl
t2A.Ld.NrInh.validation <- sample_n(t2A.Ld.NrInh, nAggl)
t2A.Ld.NrInh.model <- anti_join(t2A.Ld.NrInh, t2A.Ld.NrInh.validation, by="RLID")

#compare distribution random selection
t2A.Ld.NrInh.model <- t2A.Ld.NrInh.model %>% mutate(Model_valid = "Model")
t2A.Ld.NrInh.validation <- t2A.Ld.NrInh.validation %>% mutate(Model_valid = "Validation")
t2A.Ld.NrInh.R <- rbind(t2A.Ld.NrInh.model, t2A.Ld.NrInh.validation)
ggplot(t2A.Ld.NrInh.R, aes(x=LdSum, fill=Model_valid)) +
 geom_histogram(binwidth=20000, alpha=.45, position="identity") +
 scale_x_continuous(labels = comma_format(big.mark = ".",
              decimal.mark = ",")) +  #change scientific notation
 xlab("Lden >= 55 dB") +       #x label
 labs(fill = "Data random")           #title of legend
summary(t2A.Ld.NrInh.model$LdSum)
~~~~

~~~~
summary(t2A.Ld.NrInh.validation$LdSum)
ks.test(t2A.Ld.NrInh.model$LdSum, t2A.Ld.NrInh.validation$LdSum)
rm(tmp)

# plotting the result of Kolmogorov Smirnov
# visualization
plot(ecdf(t2A.Ld.NrInh.model$LdSum),
    xlim = range(c(t2A.Ld.NrInh.model$LdSum, t2A.Ld.NrInh.validation$LdSum)),
    col = "red")
plot(ecdf(t2A.Ld.NrInh.validation$LdSum),
    add = TRUE,
    lty = "dashed",
    col = "green")

#Plot LdSum against NurOfInhabitants
ggplot(data = t2A.Ld.NrInh.model, mapping = aes(x = log(NumberOfInhabitants) , y = log(LdSum))) +
  geom_point() +
  scale_x_continuous(labels = comma_format(big.mark = ".",
                                 decimal.mark = ",")) +
  scale_y_continuous(labels = comma_format(big.mark = ".",
                                 decimal.mark = ","))

t2A.Ld.model <- t2A.Ld.NrInh.model
t2A.Ld.validation <- t2A.Ld.NrInh.validation
```
### 2331.Regression model1

Linear model y = a + b*x
Data source: t2A.Ld.model
Look at
* heterostadicity. Residuals vs fitted
* Normal Q-Q. Follow the line
* residuals within leverage (ouside = strong influence on refression)

```{r regression.model1}

#Calculate regression
Ld.Rgr1 <- lm(LdSum ~ NumberOfInhabitants, data=t2A.Ld.model)

#Assessing model quality
glance(Ld.Rgr1) %>%   # [broom package], computes the R2, adjusted R2, sigma (RSE), AIC, BIC
  dplyr::select(adj.r.squared, sigma, AIC, BIC, p.value) #select only the statistics of interest
summary(Ld.Rgr1)

100*sigma(Ld.Rgr1)/mean(t2A.Ld.model$LdSum) #% error of SME
mean(t2A.Ld.model$LdSum) #Mean of people exposed

par(mfrow = c(2, 2))  # Split the plotting panel into a 2 x 2 grid
plot(Ld.Rgr1) + scale_x_continuous(labels = comma_format(big.mark = ".",
                                 decimal.mark = ","))
```

##### Validation model1
```{r validation.model1}
#Predict with t2A.Ld.validation
predict <- predict(Ld.Rgr1, newdata = t2A.Ld.validation, interval = "confidence")
~~~~

~~~~
Ld.Predict.1 <- cbind(t2A.Ld.validation, predict)

#Plot fitted vs data
ggplot(data = Ld.Predict.1, mapping = aes(x = LdSum, y = fit)) +
  geom_point() +
  geom_abline(intercept = 0, slope = 1) +
  scale_x_continuous(labels = comma_format(big.mark = ".",
                        decimal.mark = ",")) +
  scale_y_continuous(labels = comma_format(big.mark = ".",
                        decimal.mark = ","))

#calculate confidence interval of the sum
# sqrt[ sum(confidence_interval_individual^2) ]
Ld.Predict.1 <- Ld.Predict.1 %>%
  mutate(CI = (upr - lwr)/2) %>%
  mutate(sqr.CI = CI^2)
CI.1 = sqrt(sum(Ld.Predict.1$sqr.CI))

#Compare total values reported vs estimated + CI
sum(Ld.Predict.1$LdSum)
round(sum(Ld.Predict.1$fit), -2)
round(CI.1, -2)
```

### 2332.Regression model2
Polynomial
Linear model y = a + b*x  + c*x^2
Data source: t2A.Ld.model
Look at
* heterostadicity. Residuals vs fitted
* Normal Q-Q. Follow the line
* residuals within leverage (ouside = strong influence on refression)
```{r rgr.model2}
#Calculate regression
Ld.Rgr2 = lm(LdSum ~ poly(NumberOfInhabitants, 2, raw=TRUE), data = t2A.Ld.model)

#Assessing model quality
glance(Ld.Rgr2) %>%
  dplyr::select(adj.r.squared, sigma, AIC, BIC, p.value)
summary(Ld.Rgr2)

sigma(Ld.Rgr2)/mean(t2A.Ld.model$LdSum)
mean(t2A.Ld.model$LdSum)


par(mfrow = c(2, 2))  # Split the plotting panel into a 2 x 2 grid
plot(Ld.Rgr2) + scale_x_continuous(labels = comma_format(big.mark = ".",
                        decimal.mark = ","))
```

##### Validation model2
```{r validation.model2}
#Predict with t2A.Ld.validation
predict <- predict(Ld.Rgr2, newdata = t2A.Ld.validation, interval = "confidence")

Ld.Predict.2 <- cbind(t2A.Ld.validation, predict)
~~~~

~~~
#Plot fitted vs data
ggplot(data = Ld.Predict.2, mapping = aes(x = LdSum, y = fit)) +
  geom_point() +
  geom_abline(intercept = 0, slope = 1) +
  scale_x_continuous(labels = comma_format(big.mark = ".",
                                          decimal.mark = ",")) +
  scale_y_continuous(labels = comma_format(big.mark = ".",
                                          decimal.mark = ","))
#calculate confidence interval of the sum
# sqrt[ sum(confidence_interval_individual^2) ]
Ld.Predict.2 <- Ld.Predict.2 %>%
  mutate(CI = (upr - lwr)/2) %>%
  mutate(sqr.CI = CI^2)
CI.2 = sqrt(sum(Ld.Predict.2$sqr.CI))

#Compare total values reported vs estimated
sum(Ld.Predict.2$LdSum)
round(sum(Ld.Predict.2$fit), -2)
round(CI.2, -2)
```

### 2333.Regression model3
log-log
Linear model log(y) = a + b*log(x)
Data source: t2A.Ld.model
Look at
* heterostadicity. Residuals vs fitted
* Normal Q-Q. Follow the line
* residuals within leverage (ouside = strong influence on refression)
```{r rgr.model3}
#Calculate regression
Ld.Rgr3 = lm(log(LdSum) ~ log(NumberOfInhabitants), data = t2A.Ld.model)

#Assessing model quality
glance(Ld.Rgr3) %>%
  dplyr::select(adj.r.squared, sigma, AIC, BIC, p.value)
summary(Ld.Rgr3)

exp(sigma(Ld.Rgr3))/log(mean(t2A.Ld.model$LdSum))
log(mean(t2A.Ld.model$LdSum))


par(mfrow = c(2, 2))  # Split the plotting panel into a 2 x 2 grid
plot(Ld.Rgr3) + scale_x_continuous(labels = comma_format(big.mark = ".",
                                          decimal.mark = ","))
```

##### Validation model3
```{r validation.model3}
#Predict with t2A.Ld.validation
predict <- (predict(Ld.Rgr3, newdata = t2A.Ld.validation, interval = "confidence"))

Ld.Predict.3 <- cbind(t2A.Ld.validation, predict)

Ld.Predict.3 <- Ld.Predict.3 %>%
    mutate(fit.tr = exp(fit)) %>%    #exponential to transform log log regression
    mutate(lwr.tr = exp(lwr)) %>%
~~~

```
    mutate(upr.tr = exp(upr))

#Plot fitted vs data
ggplot(data = Ld.Predict.3, mapping = aes(x = LdSum, y = fit.tr)) +
  geom_point() +
  geom_abline(intercept = 0, slope = 1) +
  scale_x_continuous(labels = comma_format(big.mark = ".",
                                           decimal.mark = ",")) +
  scale_y_continuous(labels = comma_format(big.mark = ".",
                                           decimal.mark = ","))

#calculate confidence interval of the sum
# sqrt[ sum(confidence_interval_individual^2) ]
Ld.Predict.3 <- Ld.Predict.3 %>%
  mutate(CI = (upr.tr - lwr.tr)/2) %>%
  mutate(sqr.CI = CI^2)
CI.3 = sqrt(sum(Ld.Predict.3$sqr.CI))


#Compare total values reported vs estimated
sum(Ld.Predict.3$LdSum)
round(sum(Ld.Predict.3$fit.tr), -2)
round(CI.3, -2)
```

### 2334 Comparison of the three models
```{r models_summry}
glance(Ld.Rgr1) %>%
  dplyr::select(adj.r.squared, sigma, AIC, BIC, p.value)
glance(Ld.Rgr2) %>%
  dplyr::select(adj.r.squared, sigma, AIC, BIC, p.value)
glance(Ld.Rgr3) %>%
  dplyr::select(adj.r.squared, sigma, AIC, BIC, p.value)

summary(Ld.Rgr1)
summary(Ld.Rgr2)
summary(Ld.Rgr3)

sum(Ld.Predict.1$LdSum)
round(sum(Ld.Predict.1$fit), -2)
round((100*(sum(Ld.Predict.1$fit)-sum(Ld.Predict.1$LdSum))/sum(Ld.Predict.1$LdSum)), 1)
round(CI.1, -2)
sum(Ld.Predict.2$LdSum)
round(sum(Ld.Predict.2$fit), -2)
round(100*(sum(Ld.Predict.2$fit)-sum(Ld.Predict.2$LdSum))/sum(Ld.Predict.2$LdSum), 1)
round(CI.2, -2)
sum(Ld.Predict.3$LdSum)
round(sum(Ld.Predict.3$fit.tr), -2)
round(100*(sum(Ld.Predict.3$fit.tr)-sum(Ld.Predict.3$LdSum))/sum(Ld.Predict.3$LdSum), 1)
round(CI.3, -2)

```
```{r models_barchrt}

plot.model.Ld <- data.frame(
  name=c("reported", "model1", "model2", "model3"),
```

~~~
 Ld=c( sum(Ld.Predict.1$LdSum), sum(Ld.Predict.1$fit),
      sum(Ld.Predict.2$fit), sum(Ld.Predict.3$fit.tr)),
 Ld.low=c(0, sum(Ld.Predict.1$fit)-CI.1, sum(Ld.Predict.2$fit)-CI.2,
        sum(Ld.Predict.3$fit.tr)-CI.3),
 Ld.up=c(0, sum(Ld.Predict.1$fit)+CI.1, sum(Ld.Predict.2$fit)+CI.2,
        sum(Ld.Predict.3$fit.tr)+CI.3))

ggplot(plot.model.Ld) +
 geom_bar( aes(x=name, y=Ld), stat="identity", fill="steelblue", alpha=0.7) +
 geom_errorbar( aes(x=name, ymin=Ld.low, ymax=Ld.up), width=0.08, colour="black", alpha=0.9, size=0.8) +
 scale_y_continuous(labels = scales::comma)
```
### 2335. Estimated missing values
SPECIFY MODEL SELECTED (1,2,3, or any other). Modify number in line 1058 "predict <- (predict(Ld.Rgr2"
```{r estimate_Ld}

# Estimate LdSum: apply regression
#Gc--> Aquí s'ha de seleccionar el model que mès s'ajusta en funcio del valor estimat i variabilitat
predict <- (predict(Ld.Rgr2, newdata = t2C.Ld, interval = "confidence"))

t2C.Ld <- cbind(t2C.Ld, predict)
t2C.Ld <- t2C.Ld %>%
        mutate(LdSum = round(fit, -2)) %>%
        mutate(LdSum.error = (upr-lwr)/2)

#Average and Error for the Ld noise band mean:
LdNBpcnt <- t2.cln %>%
   filter(LdRprt == 1) %>% #Outliers not excluded
   filter(LdSum != 0) %>%  #Exclude LdSum = 0
   mutate(Ld55_pcnt = ifelse(Ld55 == -1,NA, round(100*Ld55/LdSum, 2))) %>%
   mutate(Ld60_pcnt = ifelse(Ld60 == -1,NA, round(100*Ld60/LdSum, 2))) %>%
   mutate(Ld65_pcnt = ifelse(Ld65 == -1,NA, round(100*Ld65/LdSum, 2))) %>%
   mutate(Ld70_pcnt = ifelse(Ld70 == -1,NA, round(100*Ld70/LdSum, 2))) %>%
   mutate(Ld75_pcnt = ifelse(Ld75 == -1,NA, round(100*Ld75/LdSum, 2))) %>%
   select(-18:-26, -28, -30)

Ld55.error.noout <- qt(0.975,df=length(LdNBpcnt$Ld55_pcnt[!is.na(LdNBpcnt$Ld55_pcnt)])-
1)*sd(LdNBpcnt$Ld55_pcnt, na.rm = TRUE)/sqrt(length(LdNBpcnt$Ld55_pcnt[!is.na(LdNBpcnt$Ld55_pcnt)]))
Ld60.error.noout <- qt(0.975,df=length(LdNBpcnt$Ld60_pcnt[!is.na(LdNBpcnt$Ld60_pcnt)])-
1)*sd(LdNBpcnt$Ld60_pcnt, na.rm = TRUE)/sqrt(length(LdNBpcnt$Ld60_pcnt[!is.na(LdNBpcnt$Ld60_pcnt)]))
Ld65.error.noout <- qt(0.975,df=length(LdNBpcnt$Ld65_pcnt[!is.na(LdNBpcnt$Ld65_pcnt)])-
1)*sd(LdNBpcnt$Ld65_pcnt, na.rm = TRUE)/sqrt(length(LdNBpcnt$Ld65_pcnt[!is.na(LdNBpcnt$Ld65_pcnt)]))
Ld70.error.noout <- qt(0.975,df=length(LdNBpcnt$Ld70_pcnt[!is.na(LdNBpcnt$Ld70_pcnt)])-
1)*sd(LdNBpcnt$Ld70_pcnt, na.rm = TRUE)/sqrt(length(LdNBpcnt$Ld70_pcnt[!is.na(LdNBpcnt$Ld70_pcnt)]))
Ld75.error.noout <- qt(0.975,df=length(LdNBpcnt$Ld75_pcnt[!is.na(LdNBpcnt$Ld75_pcnt)])-
1)*sd(LdNBpcnt$Ld75_pcnt, na.rm = TRUE)/sqrt(length(LdNBpcnt$Ld75_pcnt[!is.na(LdNBpcnt$Ld75_pcnt)]))
Ld55PcntMean <- mean(LdNBpcnt$Ld55_pcnt, na.rm=TRUE)
Ld60PcntMean <- mean(LdNBpcnt$Ld60_pcnt, na.rm=TRUE)
Ld65PcntMean <- mean(LdNBpcnt$Ld65_pcnt, na.rm=TRUE)
Ld70PcntMean <- mean(LdNBpcnt$Ld70_pcnt, na.rm=TRUE)
Ld75PcntMean <- mean(LdNBpcnt$Ld75_pcnt, na.rm=TRUE)

#output stats noise bands
summary(LdNBpcnt$Ld55_pcnt)
length(LdNBpcnt$Ld55_pcnt)
Ld55.error.noout
~~~

~~~~
summary(LdNBpcnt$Ld60_pcnt)
length(LdNBpcnt$Ld60_pcnt)
Ld60.error.noout

summary(LdNBpcnt$Ld65_pcnt)
length(LdNBpcnt$Ld65_pcnt)
Ld65.error.noout

summary(LdNBpcnt$Ld70_pcnt)
length(LdNBpcnt$Ld70_pcnt)
Ld70.error.noout

summary(LdNBpcnt$Ld75_pcnt)
length(LdNBpcnt$Ld75_pcnt)
Ld75.error.noout

#Estimate missing values per noise band
t2C.Ld <- t2C.Ld %>%
    mutate(Ld55 = round(LdSum * Ld55PcntMean/100, -2) ) %>%
    mutate(Ld60 = round(LdSum * Ld60PcntMean/100, -2) ) %>%
    mutate(Ld65 = round(LdSum * Ld65PcntMean/100, -2) ) %>%
    mutate(Ld70 = round(LdSum * Ld70PcntMean/100, -2)) %>%
    mutate(Ld75 = round(LdSum * Ld75PcntMean/100, -2) ) %>%
    mutate(Ld_GapFilled = ifelse(ReferenceYear < 2000 + year_t2 -2 , paste('Regression -population data', ReferenceYear), 'European average')) %>%   #create column DataSrc
    mutate(Ld_Change = 'Change')

#Calculate errors of missing values
# LdSum = NumberInhabitants * LdSum.error
# Noise bands (example Ld55):
#                    Ld55 * sqrt( (Ldsum.error/LdSum)^2  + (Ld55.error.noout/Ld55PcntMean)^2 )

t2C.Ld.errorNB <- t2C.Ld %>%
    mutate(Ld55.error = round(Ld55*sqrt((LdSum.error/LdSum)^2+(Ld55.error.noout/Ld55PcntMean)^2))) %>%
    mutate(Ld60.error = round(Ld60*sqrt((LdSum.error/LdSum)^2+(Ld60.error.noout/Ld60PcntMean)^2))) %>%
    mutate(Ld65.error = round(Ld65*sqrt((LdSum.error/LdSum)^2+(Ld65.error.noout/Ld65PcntMean)^2))) %>%
    mutate(Ld70.error = round(Ld70*sqrt((LdSum.error/LdSum)^2+(Ld70.error.noout/Ld70PcntMean)^2))) %>%
    mutate(Ld75.error = round(Ld75*sqrt((LdSum.error/LdSum)^2+(Ld75.error.noout/Ld75PcntMean)^2))) %>%
    mutate(Ld_GapFilled = ifelse(ReferenceYear < 2000 + year_t2-2 , paste('Regressione -population data', ReferenceYear), 'European average')) %>%   #create column DataSrc
    mutate(Ln_Change = 'Change')

rm(tmpA,tmpB, tmpAB)


```

### 234 Regression Lnight

 regression
  iterate for each model
    regression
    statistics of regression
    plot residuals
 plot outcome of different models
 estimate for missing data with choosen model
~~~~

~~~
#### a.Plot Lnight ~ Nr of inhabitants
 look at the scatter plot to decide alternative models
```{r plotLnPop}
#Ln Plot with outliers t2
tmp2.Ln <- t2A.Ln %>%
 filter(Ln_More100Pcnt == FALSE &
     NumberOfInhabitants > -1 &
     Outlier_all == FALSE)
 ggplot(data = t2A.Ln, mapping = aes(x = NumberOfInhabitants, y = LnSum)) +
 geom_point() +
 scale_x_continuous(labels = comma_format(big.mark = ".",
                      decimal.mark = ",")) +
 scale_y_continuous(labels = comma_format(big.mark = ".",
                      decimal.mark = ","))

#Ln Plot without outliers
tmp2.Ln <- t2A.Ln %>% filter(Ln_More100Pcnt == FALSE &
                NumberOfInhabitants > -1 &
                Outlier_all == FALSE)
ggplot(data = tmp2.Ln, mapping = aes(x = NumberOfInhabitants, y = LnSum)) +
 geom_point() +
 scale_x_continuous(labels = comma_format(big.mark = ".",
                      decimal.mark = ",")) +
 scale_y_continuous(labels = comma_format(big.mark = ".",
                      decimal.mark = ","))

t2A.Ln %>% filter(Ln_More100Pcnt == FALSE & NumberOfInhabitants > -1) %>%
               count()
t2A.Ln %>% filter(Ln_More100Pcnt == FALSE &
         NumberOfInhabitants > -1 & Outlier_all == FALSE) %>% count()
rm(tmp2.Ln)
```


#### b.Transformations

Transform LnSum to log

```{r log_trans_Ln}
#Transform if it is needed
t2A.Ln <- t2A.Ln %>%
      mutate(log.LnSum = log(LnSum))

tmp2A <- t2A.Ln %>% filter(Ln_More100Pcnt == FALSE &
                NumberOfInhabitants > -1 &
                Outlier_all == FALSE)

ggplot(data = tmp2A, mapping = aes(x = NumberOfInhabitants, y = log.LnSum)) +
 geom_point() +
 scale_x_continuous(labels = comma_format(big.mark = ".",
                      decimal.mark = ",")) +
 scale_y_continuous(labels = comma_format(big.mark = ".",
                      decimal.mark = ","))

```
~~~

~~~~
transform sqrt
```{r sqrt_trans_Ln}
#Transform if it is needed
t2A.Ln <- t2A.Ln %>%
        mutate(sqrt(LnSum))

tmp2A <- t2A.Ln %>% filter(Ln_More100Pcnt == FALSE &
                           NumberOfInhabitants > -1 &
                           Outlier_all == FALSE)

ggplot(data = t2A.Ln, mapping = aes(x = NumberOfInhabitants, y = sqrt(LnSum))) +
  geom_point() +
  scale_x_continuous(labels = comma_format(big.mark = ".",
                                           decimal.mark = ",")) +
  scale_y_continuous(labels = comma_format(big.mark = ".",
                                           decimal.mark = ","))

```


Transform loglog
```{r loglog_trans_Ln}
#Transform if it is needed
t2A.Ln <- t2A.Ln %>%
        mutate(log(NumberOfInhabitants))

tmp2A <- t2A.Ln %>% filter(Ln_More100Pcnt == FALSE &
                           NumberOfInhabitants > -1 &
                           Outlier_all == FALSE)

ggplot(data = tmp2A, mapping = aes(x = log(NumberOfInhabitants), y = log(LnSum))) +
  geom_point() +
  scale_x_continuous(labels = comma_format(big.mark = ".",
                                           decimal.mark = ",")) +
  scale_y_continuous(labels = comma_format(big.mark = ".",
                                           decimal.mark = ","))

```


#### c.Random selection of a subset for verification
Output: t2A.Ln.model to run the regression; t2A.Ln.validation to validate

Random selection filtering by range of NumberOfInhabitants

```{r rangeNrInhb_Ln}
#Select the same range of NumberOfInhabitants than the missing data

# Statistics & histogram of aggl to be gap filled
t2C.Ln %>%
  select(NumberOfInhabitants) %>%
  filter(NumberOfInhabitants > -1) %>%   #Only if NrOfInhabitants available
  summarise(n(), min(., na.rm = TRUE), max(., na.rm = TRUE), mean(NumberOfInhabitants, na.rm = TRUE) )

t2C.Ln %>%
  select(NumberOfInhabitants) %>%
  filter(NumberOfInhabitants > -1) %>%   #Only if NrOfInhabitants available
  {ggplot(., aes(x=NumberOfInhabitants)) +
  geom_histogram(binwidth=25000, fill="#69b3a2", color="#e9ecef") +
~~~~

~~~
  scale_x_continuous(labels = comma_format(big.mark = ".",
              decimal.mark = ","))}

```

INTRODUCE THE RANGE OF NUMBER OF INHABITANTS
Test Kolmogorov_Smirnov p as high as possible
```{r splitModelvalidation_Ln}
#random selection ENTER THE RANGE OF NUMBER OF INHABITANTS
#Select 70% of all aggl for regression, the rest for validation
#GC-->S'ha de marcar un rang de valors relatiu al nombre d'habitans que sigui coherent amb el rang de valors de la taula anterior (tb1_dbf) donant marge per els dos costats
#GC--> El resulat d'aquest procès ens ha de donar una p-value per sobre del 0.7. Repetir fins que això sigui així.
t2A.Ln.NrInh <- t2A.Ln %>%
  filter(Ln_More100Pcnt == FALSE) %>%
  filter(Outlier_all == FALSE) %>%
  filter(NumberOfInhabitants >= 60000 & NumberOfInhabitants <= 850000)
#ENTER THE NUMBER OF AGGLOMERATIONS
nAggl <- round(nrow(t2A.Ln.NrInh)*0.30)
nAggl
t2A.Ln.NrInh.validation <- sample_n(t2A.Ln.NrInh, nAggl)
t2A.Ln.NrInh.model <- anti_join(t2A.Ln.NrInh, t2A.Ln.NrInh.validation, by="RLID")

#compare distribution random selection
t2A.Ln.NrInh.model <- t2A.Ln.NrInh.model %>% mutate(Model_valid = "Model")
t2A.Ln.NrInh.validation <- t2A.Ln.NrInh.validation %>% mutate(Model_valid = "Validation")
t2A.Ln.NrInh.R <- rbind(t2A.Ln.NrInh.model, t2A.Ln.NrInh.validation)
ggplot(t2A.Ln.NrInh.R, aes(x=LnSum, fill=Model_valid)) +
  geom_histogram(binwidth=20000, alpha=.45, position="identity") +
  scale_x_continuous(labels = comma_format(big.mark = ".",
              decimal.mark = ",")) +  #change scientific notation
  xlab("Ln >= 50 dB") +       #x label
  labs(fill = "Data random")           #title of legend
summary(t2A.Ln.NrInh.model$LnSum)
summary(t2A.Ln.NrInh.validation$LnSum)
ks.test(t2A.Ln.NrInh.model$LnSum, t2A.Ln.NrInh.validation$LnSum)
rm(tmp)

# plotting the result of Kolmogorov Smirnov
# visualization
plot(ecdf(t2A.Ln.NrInh.model$LnSum),
     xlim = range(c(t2A.Ln.NrInh.model$LnSum, t2A.Ln.NrInh.validation$LnSum)),
     col = "red")
plot(ecdf(t2A.Ln.NrInh.validation$LnSum),
     add = TRUE,
     lty = "dashed",
     col = "green")

#Plot LnSum against NurOfInhabitants
ggplot(data = t2A.Ln.NrInh.model, mapping = aes(x = log(NumberOfInhabitants) , y = log(LnSum))) +
  geom_point() +
  scale_x_continuous(labels = comma_format(big.mark = ".",
                          decimal.mark = ",")) +
  scale_y_continuous(labels = comma_format(big.mark = ".",
                          decimal.mark = ","))

t2A.Ln.model <- t2A.Ln.NrInh.model
~~~

~~~~
t2A.Ln.validation <- t2A.Ln.NrInh.validation
```
### 2341.Regression model1

Linear model y = a + b*x
Data source: t2A.Ln.model
Look at
* heterostadicity. Residuals vs fitted
* Normal Q-Q. Follow the line
* residuals within leverage (ouside = strong influence on refression)

```{r regression.model1_Ln}

#Calculate regression
Ln.Rgr1 <- lm(LnSum ~ NumberOfInhabitants, data=t2A.Ln.model)

#Assessing model quality
glance(Ln.Rgr1) %>%   # [broom package], computes the R2, adjusted R2, sigma (RSE), AIC, BIC
  dplyr::select(adj.r.squared, sigma, AIC, BIC, p.value) #select only the statistics of interest
summary(Ln.Rgr1)

100*sigma(Ln.Rgr1)/mean(t2A.Ln.model$LnSum) #% error of SME
mean(t2A.Ln.model$LnSum) #Mean of people exposed

par(mfrow = c(2, 2))  # Split the plotting panel into a 2 x 2 grid
plot(Ln.Rgr1) + scale_x_continuous(labels = comma_format(big.mark = ".",
                                            decimal.mark = ","))
```

##### Validation model1
```{r validation.model1_Ln}
#Predict with t2A.Ln.validation
predict <- predict(Ln.Rgr1, newdata = t2A.Ln.validation, interval = "confidence")

Ln.Predict.1 <- cbind(t2A.Ln.validation, predict)

#Plot fitted vs data
ggplot(data = Ln.Predict.1, mapping = aes(x = LnSum, y = fit)) +
  geom_point() +
  geom_abline(intercept = 0, slope = 1) +
  scale_x_continuous(labels = comma_format(big.mark = ".",
                                            decimal.mark = ",")) +
  scale_y_continuous(labels = comma_format(big.mark = ".",
                                            decimal.mark = ","))

#calculate confidence interval of the sum
# sqrt[ sum(confidence_interval_individual^2) ]
Ln.Predict.1 <- Ln.Predict.1 %>%
  mutate(CI = (upr - lwr)/2) %>%
  mutate(sqr.CI = CI^2)
CI.1 = sqrt(sum(Ln.Predict.1$sqr.CI))

#Compare total values reported vs estimated + CI
sum(Ln.Predict.1$LnSum)
round(sum(Ln.Predict.1$fit), -2)
round(CI.1, -2)
```
~~~~

~~~
### 2342.Regression model2
Polynomial
Linear model y = a + b*x  + c*x^2
Data source: t2A.Ln.model
Look at
* heterostadicity. Residuals vs fitted
* Normal Q-Q. Follow the line
* residuals within leverage (ouside = strong influence on refression)
```{r rgr.model2_Ln}
#Calculate regression
Ln.Rgr2 = lm(LnSum ~ poly(NumberOfInhabitants, 2, raw=TRUE), data = t2A.Ln.model)

#Assessing model quality
glance(Ln.Rgr2) %>%
  dplyr::select(adj.r.squared, sigma, AIC, BIC, p.value)
summary(Ln.Rgr2)

sigma(Ln.Rgr2)/mean(t2A.Ln.model$LnSum)
mean(t2A.Ln.model$LnSum)


par(mfrow = c(2, 2))  # Split the plotting panel into a 2 x 2 grid
plot(Ln.Rgr2) + scale_x_continuous(labels = comma_format(big.mark = ".",
                                   decimal.mark = ","))
~~~

##### Validation model2
```{r validation.model2_Ln}
#Predict with t2A.Ln.validation
predict <- predict(Ln.Rgr2, newdata = t2A.Ln.validation, interval = "confidence")

Ln.Predict.2 <- cbind(t2A.Ln.validation, predict)

#Plot fitted vs data
ggplot(data = Ln.Predict.2, mapping = aes(x = LnSum, y = fit)) +
  geom_point() +
  geom_abline(intercept = 0, slope = 1) +
  scale_x_continuous(labels = comma_format(big.mark = ".",
                                   decimal.mark = ",")) +
  scale_y_continuous(labels = comma_format(big.mark = ".",
                                   decimal.mark = ","))
#calculate confidence interval of the sum
# sqrt[ sum(confidence_interval_individual^2) ]
Ln.Predict.2 <- Ln.Predict.2 %>%
  mutate(CI = (upr - lwr)/2) %>%
  mutate(sqr.CI = CI^2)
CI.2 = sqrt(sum(Ln.Predict.2$sqr.CI))

#Compare total values reported vs estimated
sum(Ln.Predict.2$LnSum)
round(sum(Ln.Predict.2$fit), -2)
round(CI.2, -2)
```

### 2333.Regression model3
log-log
```
```

~~~~
Linear model log(y) = a + b*log(x)
Data source: t2A.Ln.model
Look at
* heterostadicity. Residuals vs fitted
* Normal Q-Q. Follow the line
* residuals within leverage (ouside = strong influence on refression)
```{r rgr.model3_Ln}
#Calculate regression
Ln.Rgr3 = lm(log(LnSum) ~ log(NumberOfInhabitants), data = t2A.Ln.model)

#Assessing model quality
glance(Ln.Rgr3) %>%
  dplyr::select(adj.r.squared, sigma, AIC, BIC, p.value)
summary(Ln.Rgr3)

exp(sigma(Ln.Rgr3))/log(mean(t2A.Ln.model$LnSum))
log(mean(t2A.Ln.model$LnSum))


par(mfrow = c(2, 2))  # Split the plotting panel into a 2 x 2 grid
plot(Ln.Rgr3) + scale_x_continuous(labels = comma_format(big.mark = ".",
                                            decimal.mark = ","))
```

##### Validation model3
```{r validation.model3_Ln}
#Predict with t2A.Ln.validation
predict <- (predict(Ln.Rgr3, newdata = t2A.Ln.validation, interval = "confidence"))

Ln.Predict.3 <- cbind(t2A.Ln.validation, predict)

Ln.Predict.3 <- Ln.Predict.3 %>%
        mutate(fit.tr = exp(fit)) %>%    #exponential to transform log log regression
        mutate(lwr.tr = exp(lwr)) %>%
        mutate(upr.tr = exp(upr))

#Plot fitted vs data
ggplot(data = Ln.Predict.3, mapping = aes(x = LnSum, y = fit.tr)) +
  geom_point() +
  geom_abline(intercept = 0, slope = 1) +
  scale_x_continuous(labels = comma_format(big.mark = ".",
                                            decimal.mark = ",")) +
  scale_y_continuous(labels = comma_format(big.mark = ".",
                                            decimal.mark = ","))

#calculate confidence interval of the sum
# sqrt[ sum(confidence_interval_individual^2) ]
Ln.Predict.3 <- Ln.Predict.3 %>%
  mutate(CI = (upr.tr - lwr.tr)/2) %>%
  mutate(sqr.CI = CI^2)
CI.3 = sqrt(sum(Ln.Predict.3$sqr.CI))


#Compare total values reported vs estimated
sum(Ln.Predict.3$LnSum)
round(sum(Ln.Predict.3$fit.tr), -2)
round(CI.3, -2)
~~~~

```
```

### 2334 Comparison of the three models
```{r models_summry_Ln}
glance(Ln.Rgr1) %>%
  dplyr::select(adj.r.squared, sigma, AIC, BIC, p.value)
glance(Ln.Rgr2) %>%
  dplyr::select(adj.r.squared, sigma, AIC, BIC, p.value)
glance(Ln.Rgr3) %>%
  dplyr::select(adj.r.squared, sigma, AIC, BIC, p.value)

summary(Ln.Rgr1)
summary(Ln.Rgr2)
summary(Ln.Rgr3)

sum(Ln.Predict.1$LnSum)
round(sum(Ln.Predict.1$fit), -2)
round((100*(sum(Ln.Predict.1$fit)-sum(Ln.Predict.1$LnSum))/sum(Ln.Predict.1$LnSum)), 1)
round(CI.1, -2)
sum(Ln.Predict.2$LnSum)
round(sum(Ln.Predict.2$fit), -2)
round(100*(sum(Ln.Predict.2$fit)-sum(Ln.Predict.2$LnSum))/sum(Ln.Predict.2$LnSum), 1)
round(CI.2, -2)
sum(Ln.Predict.3$LnSum)
round(sum(Ln.Predict.3$fit.tr), -2)
round(100*(sum(Ln.Predict.3$fit.tr)-sum(Ln.Predict.3$LnSum))/sum(Ln.Predict.3$LnSum), 1)
round(CI.3, -2)

```

```{r models_barchrt_Ln}

plot.model.Ln <- data.frame(
  name=c("reported", "model1", "model2", "model3"),
  Ln=c( sum(Ln.Predict.1$LnSum), sum(Ln.Predict.1$fit),
        sum(Ln.Predict.2$fit), sum(Ln.Predict.3$fit.tr)),
  Ln.low=c(0, sum(Ln.Predict.1$fit)-CI.1, sum(Ln.Predict.2$fit)-CI.2,
          sum(Ln.Predict.3$fit.tr)-CI.3),
  Ln.up=c(0, sum(Ln.Predict.1$fit)+CI.1, sum(Ln.Predict.2$fit)+CI.2,
          sum(Ln.Predict.3$fit.tr)+CI.3))

ggplot(plot.model.Ln) +
  geom_bar( aes(x=name, y=Ln), stat="identity", fill="steelblue", alpha=0.7) +
  geom_errorbar( aes(x=name, ymin=Ln.low, ymax=Ln.up), width=0.08, colour="black", alpha=0.9, size=0.8) +
  scale_y_continuous(labels = scales::comma)
```

### 2335. Estimated missing values
SPECIFY MODEL SELECTED (1,2,3, or any other). Modify number in line 1058 "predict <- (predict(Ln.Rgr2"
```{r estimate_Ln}

# Estimate LnSum: apply regression
#Gc--> Aquí s'ha de seleccionar el model que mès s'ajusta en funcio del valor estimat i variabilitat
predict <- (predict(Ln.Rgr1, newdata = t2C.Ln, interval = "confidence"))
```

```
t2C.Ln <- cbind(t2C.Ln, predict)
t2C.Ln <- t2C.Ln %>%
        mutate(LnSum = round(fit, -2)) %>%
        mutate(LnSum.error = (upr-lwr)/2)

#Average and Error for the Ln noise band mean:
LnNBpcnt <- t2.cln %>%
   filter(LnRprt == 1) %>% #Outliers not excluded
   filter(LnSum != 0) %>%  #Exclude LnSum = 0
   mutate(Ln50_pcnt = ifelse(Ln50 == -1,NA, round(100*Ln50/LnSum, 2))) %>%
   mutate(Ln55_pcnt = ifelse(Ln55 == -1,NA, round(100*Ln55/LnSum, 2))) %>%
   mutate(Ln60_pcnt = ifelse(Ln60 == -1,NA, round(100*Ln60/LnSum, 2))) %>%
   mutate(Ln65_pcnt = ifelse(Ln65 == -1,NA, round(100*Ln65/LnSum, 2))) %>%
   mutate(Ln70_pcnt = ifelse(Ln70 == -1,NA, round(100*Ln70/LnSum, 2))) %>%
   select(-18:-26, -28, -30)

Ln50.error.noout <- qt(0.975,df=length(LnNBpcnt$Ln50_pcnt[!is.na(LnNBpcnt$Ln50_pcnt)])-
1)*sd(LnNBpcnt$Ln50_pcnt, na.rm = TRUE)/sqrt(length(LnNBpcnt$Ln50_pcnt[!is.na(LnNBpcnt$Ln50_pcnt)]))
Ln55.error.noout <- qt(0.975,df=length(LnNBpcnt$Ln55_pcnt[!is.na(LnNBpcnt$Ln55_pcnt)])-
1)*sd(LnNBpcnt$Ln55_pcnt, na.rm = TRUE)/sqrt(length(LnNBpcnt$Ln55_pcnt[!is.na(LnNBpcnt$Ln55_pcnt)]))
Ln60.error.noout <- qt(0.975,df=length(LnNBpcnt$Ln60_pcnt[!is.na(LnNBpcnt$Ln60_pcnt)])-
1)*sd(LnNBpcnt$Ln60_pcnt, na.rm = TRUE)/sqrt(length(LnNBpcnt$Ln60_pcnt[!is.na(LnNBpcnt$Ln60_pcnt)]))
Ln65.error.noout <- qt(0.975,df=length(LnNBpcnt$Ln65_pcnt[!is.na(LnNBpcnt$Ln65_pcnt)])-
1)*sd(LnNBpcnt$Ln65_pcnt, na.rm = TRUE)/sqrt(length(LnNBpcnt$Ln65_pcnt[!is.na(LnNBpcnt$Ln65_pcnt)]))
Ln70.error.noout <- qt(0.975,df=length(LnNBpcnt$Ln70_pcnt[!is.na(LnNBpcnt$Ln70_pcnt)])-
1)*sd(LnNBpcnt$Ln70_pcnt, na.rm = TRUE)/sqrt(length(LnNBpcnt$Ln70_pcnt[!is.na(LnNBpcnt$Ln70_pcnt)]))

Ln50PcntMean <- mean(LnNBpcnt$Ln50_pcnt, na.rm=TRUE)
Ln55PcntMean <- mean(LnNBpcnt$Ln55_pcnt, na.rm=TRUE)
Ln60PcntMean <- mean(LnNBpcnt$Ln60_pcnt, na.rm=TRUE)
Ln65PcntMean <- mean(LnNBpcnt$Ln65_pcnt, na.rm=TRUE)
Ln70PcntMean <- mean(LnNBpcnt$Ln70_pcnt, na.rm=TRUE)

#output stats noise bands
summary(LnNBpcnt$Ln50_pcnt)
length(LnNBpcnt$Ln50_pcnt)
Ln50.error.noout

summary(LnNBpcnt$Ln55_pcnt)
length(LnNBpcnt$Ln55_pcnt)
Ln55.error.noout

summary(LnNBpcnt$Ln60_pcnt)
length(LnNBpcnt$Ln60_pcnt)
Ln60.error.noout

summary(LnNBpcnt$Ln65_pcnt)
length(LnNBpcnt$Ln65_pcnt)
Ln65.error.noout

summary(LnNBpcnt$Ln70_pcnt)
length(LnNBpcnt$Ln70_pcnt)
Ln70.error.noout

#Estimate missing values per noise band
t2C.Ln <- t2C.Ln %>%
      mutate(Ln50 = round(LnSum * Ln50PcntMean/100, -2) ) %>%
```

~~~
    mutate(Ln55 = round(LnSum * Ln55PcntMean/100, -2) ) %>%
    mutate(Ln60 = round(LnSum * Ln60PcntMean/100, -2) ) %>%
    mutate(Ln65 = round(LnSum * Ln65PcntMean/100, -2) ) %>%
    mutate(Ln70 = round(LnSum * Ln70PcntMean/100, -2)) %>%
    mutate(Ln_GapFilled = ifelse(ReferenceYear < 2000 + year_t2 -2 , paste('Regression -population data',
ReferenceYear), 'European average')) %>%  #create column DataSrc
    mutate(Ln_Change = 'Change')

#Calculate errors of missing values
# LnSum = NumberInhabitants * LnSum.error
# Noise bands (example Ln55):
#                Ln55 * sqrt( (Lnsum.error/LnSum)^2  + (Ln55.error.noout/Ln55PcntMean)^2 )

t2C.Ln.errorNB <- t2C.Ln %>%
    mutate(Ln50.error = round(Ln50*sqrt((LnSum.error/LnSum)^2+(Ln50.error.noout/Ln50PcntMean)^2))) %>%
    mutate(Ln55.error = round(Ln55*sqrt((LnSum.error/LnSum)^2+(Ln55.error.noout/Ln55PcntMean)^2))) %>%
    mutate(Ln60.error = round(Ln60*sqrt((LnSum.error/LnSum)^2+(Ln60.error.noout/Ln60PcntMean)^2))) %>%
    mutate(Ln65.error = round(Ln65*sqrt((LnSum.error/LnSum)^2+(Ln65.error.noout/Ln65PcntMean)^2))) %>%
    mutate(Ln70.error = round(Ln70*sqrt((LnSum.error/LnSum)^2+(Ln70.error.noout/Ln70PcntMean)^2))) %>%
    mutate(Ln_GapFilled = ifelse(ReferenceYear < 2000 + year_t2-2 , paste('Regressione -population data',
ReferenceYear), 'European average')) %>%  #create column DataSrc
    mutate(Ln_Change = 'Change')
```

## 2.4 Merge all data and calculate European aggregates
```{r  A+B+C}

#Aggregate A+B+C for Lden and Lnight-------------------
tmp2ALd <- t2A.Ld %>%
    select(2:14, 18, 19)
tmp2BLd <- t2B.Ld %>%
    select(2:14, 18, 19)
tmp2CLd <- t2C.Ld %>%
    select(2:14, 22, 23)
t2.Ld.ABC <- rbind(tmp2ALd, tmp2BLd, tmp2CLd)


tmp2ALn <- t2A.Ln %>%
    select(2:14, 18, 19)
tmp2BLn <- t2B.Ln %>%
    select(2:14, 18, 19)
tmp2CLn <- t2C.Ln %>%
    select(2:14, 22, 23)
t2.Ln.ABC <- rbind(tmp2ALn, tmp2BLn, tmp2CLn) %>%
    select(-3, -5:-7)


#Merge Lden and Lnight in the same table
t2.ABC <- merge(t2.Ld.ABC, t2.Ln.ABC, by=c('Ctry', 'Ctry2', 'RLID'))
rm(tmp2ALd, tmp2BLd, tmp2CLd, t2.Ld.ABC, tmp2ALn,tmp2BLn, tmp2CLn, t2.Ln.ABC )

#Add additional columns removed at the beginning
#Select columns from t2 to be added
tmp.t2 <- t2 %>%
    select(1:2, 4:6, 9, 11, 13, 16:19, 105)
#Merge and reorder
t2.ABC <- merge(t2.ABC, tmp.t2, by=c('Ctry', 'Ctry2', 'RLID'))
~~~

```
#t2.ABC <- t2.ABC[, c(1,2, 27:28, 5, 3, 6, 30, 8, 31, 29, 32, 7, 33:36, 9:14, 18:23, 16:17, 25:26 )] #Reorder columns
rm(tmp.t2)

#Aggregate errors--------------------------------
tmp2Ld.error <- t2C.Ld.errorNB %>%
    select(2, 3, 5:7, 24:28, 21)
tmp2Ln.error <- t2C.Ln.errorNB %>%
    select(2, 3, 5:7, 24:28, 21)

t2.error <- full_join(tmp2Ld.error, tmp2Ln.error, by=c('Ctry', 'Ctry2', 'RLID', 'AggloNameEn', 'EU28'))

t2.error <- t2.error %>% #Calculate error squares needed for totals
    mutate(Ld55sq = Ld55.error^2) %>%
    mutate(Ld60sq = Ld60.error^2) %>%
    mutate(Ld65sq = Ld65.error^2) %>%
    mutate(Ld70sq = Ld70.error^2) %>%
    mutate(Ld75sq = Ld75.error^2) %>%
    mutate(LdSumsq = LdSum.error^2) %>%
    mutate(Ln50sq = Ln50.error^2) %>%
    mutate(Ln55sq = Ln55.error^2) %>%
    mutate(Ln60sq = Ln60.error^2) %>%
    mutate(Ln65sq = Ln65.error^2) %>%
    mutate(Ln70sq = Ln70.error^2) %>%
    mutate(LnSumsq = LnSum.error^2) %>%
    mutate(LdSum.error = round(LdSum.error, -2)) %>% #After calculating the quares of the errors we round
individual errors to the nearest hundred
    mutate(Ld55.error = round(Ld55.error, -2)) %>%
    mutate(Ld60.error = round(Ld60.error, -2)) %>%
    mutate(Ld65.error = round(Ld65.error, -2)) %>%
    mutate(Ld70.error = round(Ld70.error, -2)) %>%
    mutate(Ld75.error = round(Ld75.error, -2)) %>%
    mutate(LnSum.error = round(LnSum.error, -2)) %>%
    mutate(Ln50.error = round(Ln50.error, -2)) %>%
    mutate(Ln55.error = round(Ln55.error, -2)) %>%
    mutate(Ln60.error = round(Ln60.error, -2)) %>%
    mutate(Ln65.error = round(Ln65.error, -2)) %>%
    mutate(Ln70.error = round(Ln70.error, -2))

#Calculate aggregated figures for Europe with corresponding errors----------------
# Lden
t2.Ld.EU27 <- t2.ABC %>%
 filter(EU28 == 'Yes') %>%
 filter(Ctry != 'GB') %>%
 mutate(Ld55 = ifelse(Ld55 > -1, Ld55, NA)) %>%
 mutate(Ld60 = ifelse(Ld60 > -1, Ld60, NA)) %>%
 mutate(Ld65 = ifelse(Ld65 > -1, Ld65, NA)) %>%
 mutate(Ld70 = ifelse(Ld70 > -1, Ld70, NA)) %>%
 mutate(Ld75 = ifelse(Ld75 > -1, Ld75, NA)) %>%
 mutate(LdSum = ifelse(LdSum > -1, LdSum, NA)) %>%
 summarise(across(Ld55:LdSum, sum, na.rm = TRUE)) %>%
 mutate(Ctry = 'EU27')

t2.Ld.EU27UK <- t2.ABC %>%
 filter(EU28 == 'Yes') %>%
 mutate(Ld55 = ifelse(Ld55 > -1, Ld55, NA)) %>%
 mutate(Ld60 = ifelse(Ld60 > -1, Ld60, NA)) %>%
 mutate(Ld65 = ifelse(Ld65 > -1, Ld65, NA)) %>%
 mutate(Ld70 = ifelse(Ld70 > -1, Ld70, NA)) %>%
 mutate(Ld75 = ifelse(Ld75 > -1, Ld75, NA)) %>%
 mutate(LdSum = ifelse(LdSum > -1, LdSum, NA)) %>%
 summarise(across(Ld55:LdSum, sum, na.rm = TRUE)) %>%
 mutate(Ctry = 'EU27 and UK')

t2.Ld.EEA32UK <- t2.ABC %>%
 mutate(Ld55 = ifelse(Ld55 > -1, Ld55, NA)) %>%
 mutate(Ld60 = ifelse(Ld60 > -1, Ld60, NA)) %>%
 mutate(Ld65 = ifelse(Ld65 > -1, Ld65, NA)) %>%
 mutate(Ld70 = ifelse(Ld70 > -1, Ld70, NA)) %>%
 mutate(Ld75 = ifelse(Ld75 > -1, Ld75, NA)) %>%
 mutate(LdSum = ifelse(LdSum > -1, LdSum, NA)) %>%
 summarise(across(Ld55:LdSum, sum, na.rm = TRUE)) %>%
 mutate(Ctry = 'EEA32 and UK -Turkey not Included')

t2.Ld.aggr <- rbind(t2.Ld.EU27, t2.Ld.EU27UK, t2.Ld.EEA32UK)

#Lden errors
t2.Ld.EU27.error <- t2.error %>%
 filter(EU28 == 'Yes') %>%
 filter(Ctry != 'GB') %>%
 summarise_at(vars(Ld55sq:LdSumsq), sum, na.rm = TRUE) %>%
 mutate(Ctry = 'EU27')

t2.Ld.EU27UK.error <- t2.error %>%
 filter(EU28 == 'Yes') %>%
 summarise_at(vars(Ld55sq:LdSumsq), sum, na.rm = TRUE) %>%
 mutate(Ctry = 'EU27 and UK')

t2.Ld.EEA32UK.error <- t2.error %>%
 summarise_at(vars(Ld55sq:LdSumsq), sum, na.rm = TRUE) %>%
 mutate(Ctry = 'EEA32 and UK -Turkey not Included')

t2.Ld.aggr.error <- rbind(t2.Ld.EU27.error, t2.Ld.EU27UK.error, t2.Ld.EEA32UK.error)

t2.Ld.aggr.error <- t2.Ld.aggr.error %>%
    mutate(Ld55.error = round(sqrt(Ld55sq), -2)) %>%
    mutate(Ld60.error =  round(sqrt(Ld60sq), -2)) %>%
    mutate(Ld65.error =  round(sqrt(Ld65sq), -2)) %>%
    mutate(Ld70.error =  round(sqrt(Ld70sq), -2)) %>%
    mutate(Ld75.error =  round(sqrt(Ld75sq), -2)) %>%
    mutate(LdSum.error =  round(sqrt(LdSumsq), -2))

# Lnight
t2.Ln.EU27 <- t2.ABC %>%
 filter(EU28 == 'Yes') %>%
 filter(Ctry != 'GB') %>%
 mutate(Ln50 = ifelse(Ln50 > -1, Ln50, NA)) %>%
 mutate(Ln55 = ifelse(Ln55 > -1, Ln55, NA)) %>%
 mutate(Ln60 = ifelse(Ln60 > -1, Ln60, NA)) %>%
 mutate(Ln65 = ifelse(Ln65 > -1, Ln65, NA)) %>%
 mutate(Ln70 = ifelse(Ln70 > -1, Ln70, NA)) %>%
 mutate(LnSum = ifelse(LnSum > -1, LnSum, NA)) %>%
 summarise(across(Ln50:LnSum, sum, na.rm = TRUE)) %>%
 mutate(Ctry = 'EU27')
```

```
t2.Ln.EU27UK <- t2.ABC %>%
 filter(EU28 == 'Yes') %>%
 mutate(Ln50 = ifelse(Ln50 > -1, Ln50, NA)) %>%
 mutate(Ln55 = ifelse(Ln55 > -1, Ln55, NA)) %>%
 mutate(Ln60 = ifelse(Ln60 > -1, Ln60, NA)) %>%
 mutate(Ln65 = ifelse(Ln65 > -1, Ln65, NA)) %>%
 mutate(Ln70 = ifelse(Ln70 > -1, Ln70, NA)) %>%
 mutate(LnSum = ifelse(LnSum > -1, LnSum, NA)) %>%
 summarise(across(Ln50:LnSum, sum, na.rm = TRUE)) %>%
 mutate(Ctry = 'EU27 and UK')

t2.Ln.EEA32UK <- t2.ABC %>%
 mutate(Ln50 = ifelse(Ln60 > -1, Ln50, NA)) %>%
 mutate(Ln55 = ifelse(Ln55 > -1, Ln55, NA)) %>%
 mutate(Ln60 = ifelse(Ln60 > -1, Ln60, NA)) %>%
 mutate(Ln65 = ifelse(Ln65 > -1, Ln65, NA)) %>%
 mutate(Ln70 = ifelse(Ln70 > -1, Ln70, NA)) %>%
 mutate(LnSum = ifelse(LnSum > -1, LnSum, NA)) %>%
 summarise(across(Ln50:LnSum, sum, na.rm = TRUE)) %>%
 mutate(Ctry = 'EEA32 and UK -Turkey not Included')

t2.Ln.aggr <- rbind(t2.Ln.EU27, t2.Ln.EU27UK, t2.Ln.EEA32UK)

#Lnight errors
t2.Ln.EU27.error <- t2.error %>%
 filter(EU28 == 'Yes') %>%
 filter(Ctry != 'GB') %>%
 summarise_at(vars(Ln50sq:LnSumsq), sum, na.rm = TRUE) %>%
 mutate(Ctry = 'EU27')

t2.Ln.EU27UK.error <- t2.error %>%
 filter(EU28 == 'Yes') %>%
 summarise_at(vars(Ln50sq:LnSumsq), sum, na.rm = TRUE) %>%
 mutate(Ctry = 'EU27 and UK')

t2.Ln.EEA32UK.error <- t2.error %>%
 summarise_at(vars(Ln50sq:LnSumsq), sum, na.rm = TRUE) %>%
 mutate(Ctry = 'EEA32 and UK -Turkey not Included')

t2.Ln.aggr.error <- rbind(t2.Ln.EU27.error, t2.Ln.EU27UK.error, t2.Ln.EEA32UK.error)

t2.Ln.aggr.error <- t2.Ln.aggr.error %>%
   mutate(Ln50.error =  round(sqrt(Ln50sq), -2)) %>%
   mutate(Ln55.error = round(sqrt(Ln55sq), -2)) %>%
   mutate(Ln60.error =  round(sqrt(Ln60sq), -2)) %>%
   mutate(Ln65.error =  round(sqrt(Ln65sq), -2)) %>%
   mutate(Ln70.error =  round(sqrt(Ln70sq), -2)) %>%
   mutate(LnSum.error =  round(sqrt(LnSumsq), -2))

#Join aggregate figures for Europe Lden & Lnight--------------------
t2.aggregated <- join(t2.Ld.aggr, t2.Ln.aggr)

t2.aggregated.error <- join(t2.Ld.aggr.error, t2.Ln.aggr.error) %>%
       select(7:13, 20:25)

#Add aggregated to general table
```

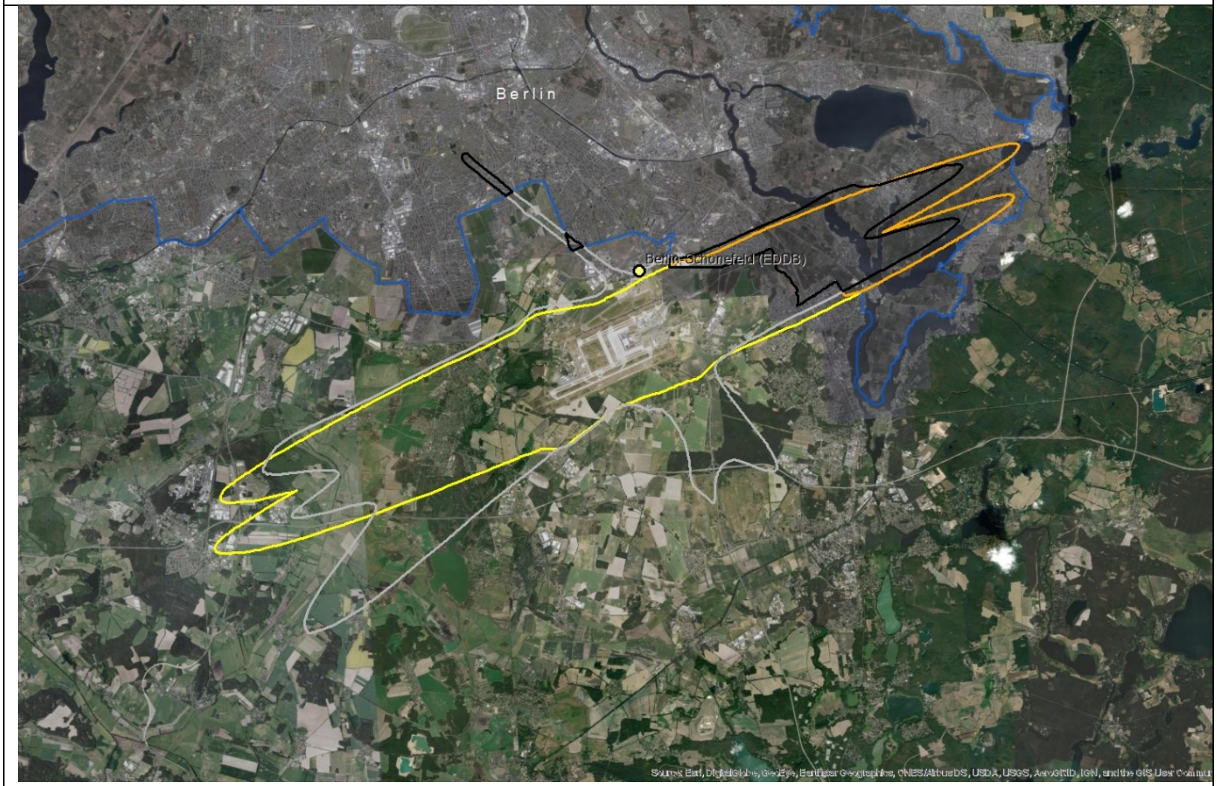
~~~
t2.export <- bind_rows(t2.ABC, t2.aggregated)
t2.export.error <- bind_rows(t2.error, t2.aggregated.error) %>%
       select(1:17)

#Export------
fn_out = paste("Agg_", nsrc, '_20', year_t2, '.xlsx', sep="")  #File name output
write.xlsx(t2.export,(here::here("output", fn_out)), sheetName = nsrc)

fn_out = paste('Agg_',  nsrc, '_20', year_t2, '_error.xlsx', sep="")
write.xlsx(t2.export.error,(here::here("output", fn_out)))
```
~~~

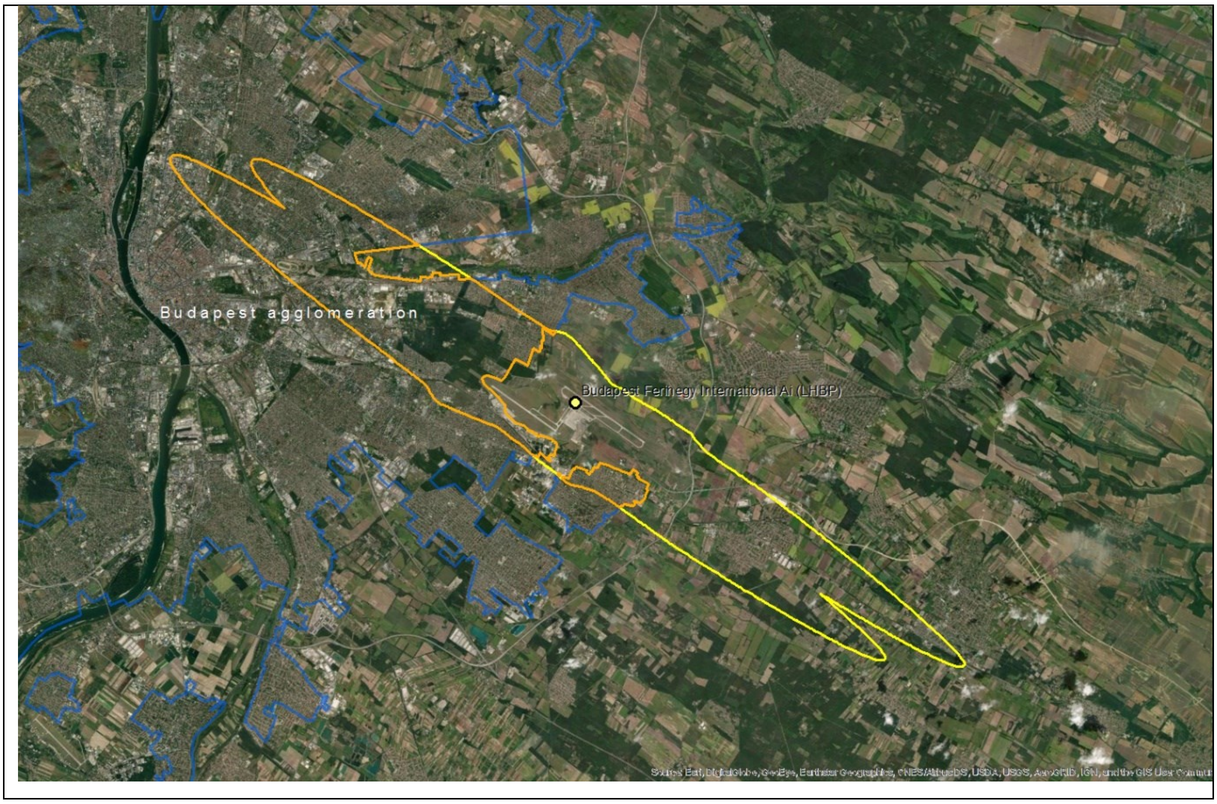
Annex 3
Reported and delineated contour for L_{den} 55 dB (major airports)

This annex provides an illustration of the reported contours for L_{den} 55 dB from major airports, and the estimated delineation according to the method described in this report.

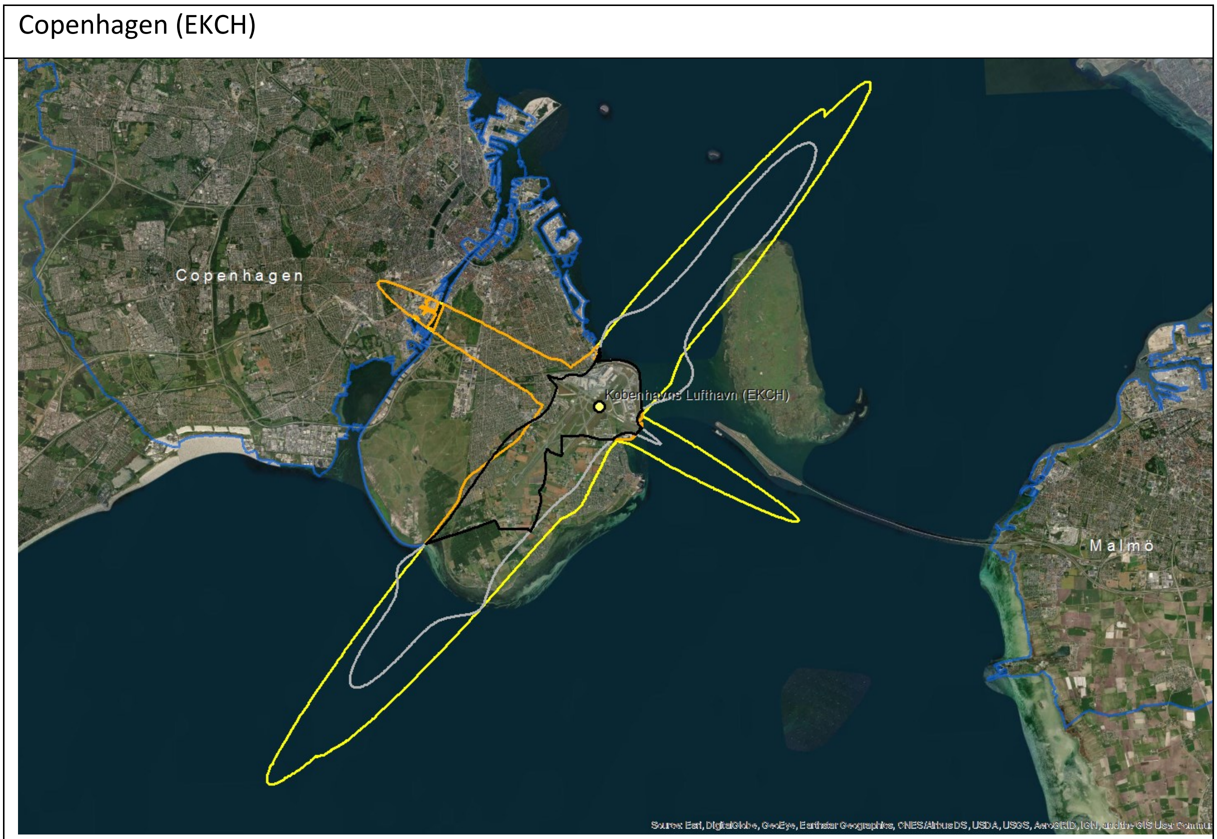
Legend:

- Delineation of the **agglomeration**: blue line
- **Reported** contour :
  - Contour outside the agglomeration: Light gray
  - Contour inside the agglomeration: black
- **Estimated** contour:
  - Contour outside the agglomeration: yellow
  - Contour inside the agglomeration: orange

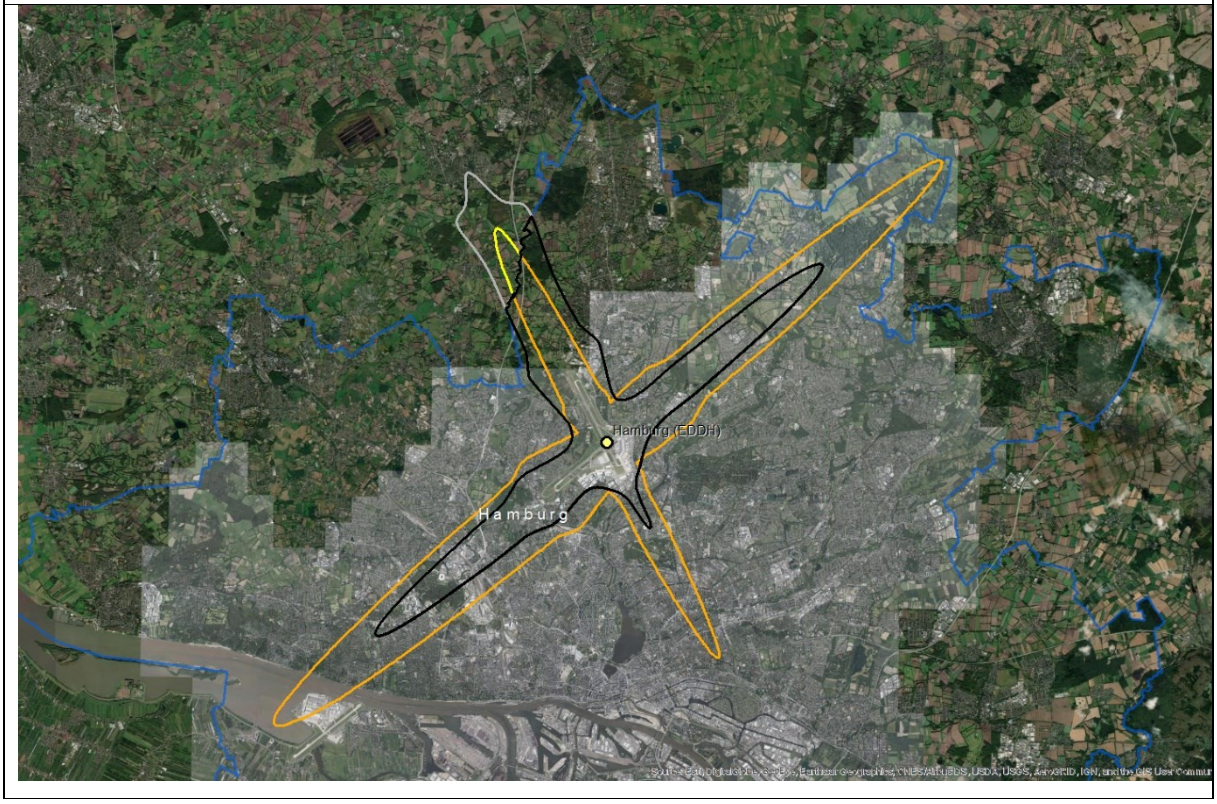
Berlin Schonefeld (EDDB)
 Source: Esri, DigitalGlobe, GeoEye, Earthstar Geographics, CNES/Airbus DS, USDA, USGS, AeroGRID, IGN, and the GIS User Commu
Budapest (LHBP)


Budapest agglomeration
Budapest Ferihegy International Ai (LHBP)
Source: Esri, DigitalGlobe, GeoEye, Earthstar Geographics, CNES/Airbus DS, USDA, USGS, AeroGRID, IGN, and the GIS User Commun

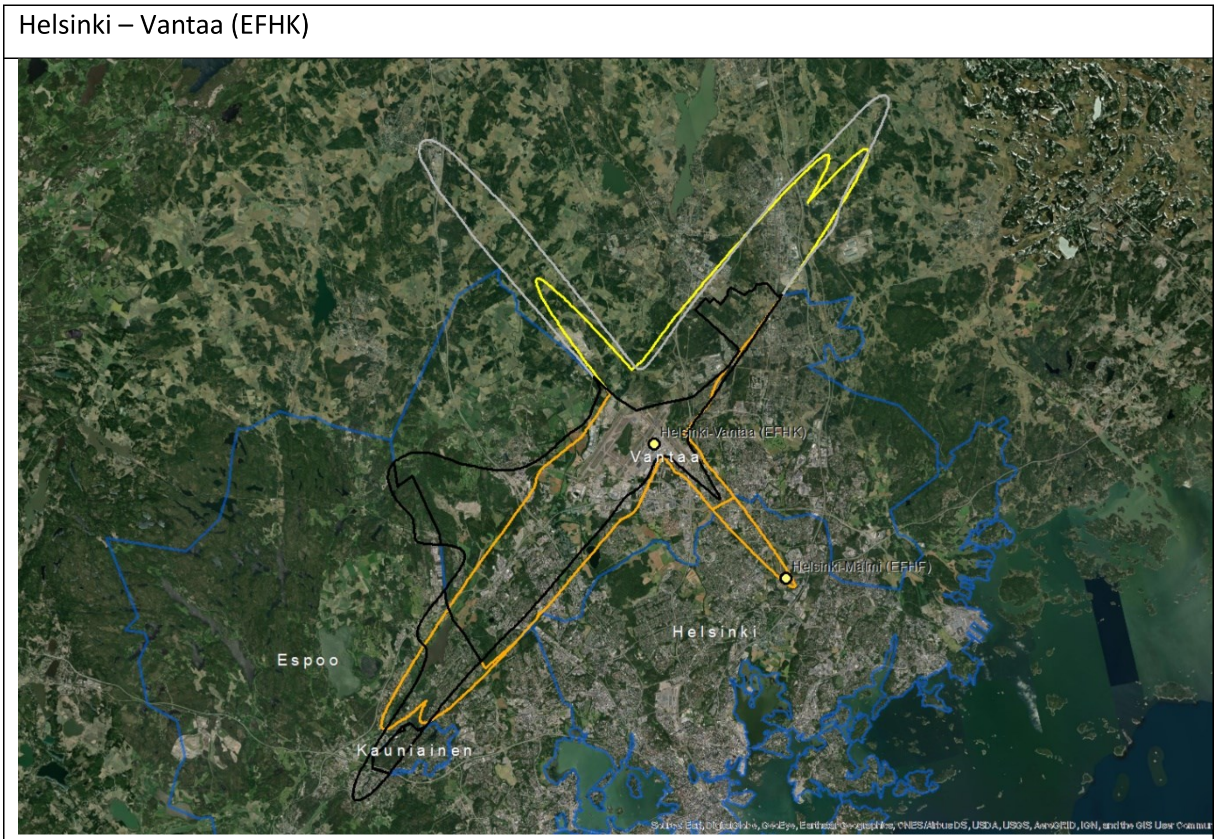
Copenhagen (EKCH)


Copenhagen
Københavns Lufthavn (EKCH)
Malmö
Source: Esri, DigitalGlobe, GeoEye, Earthstar Geographics, CNES/Airbus DS, USDA, USGS, AeroGRID, IGN, and the GIS User Commun

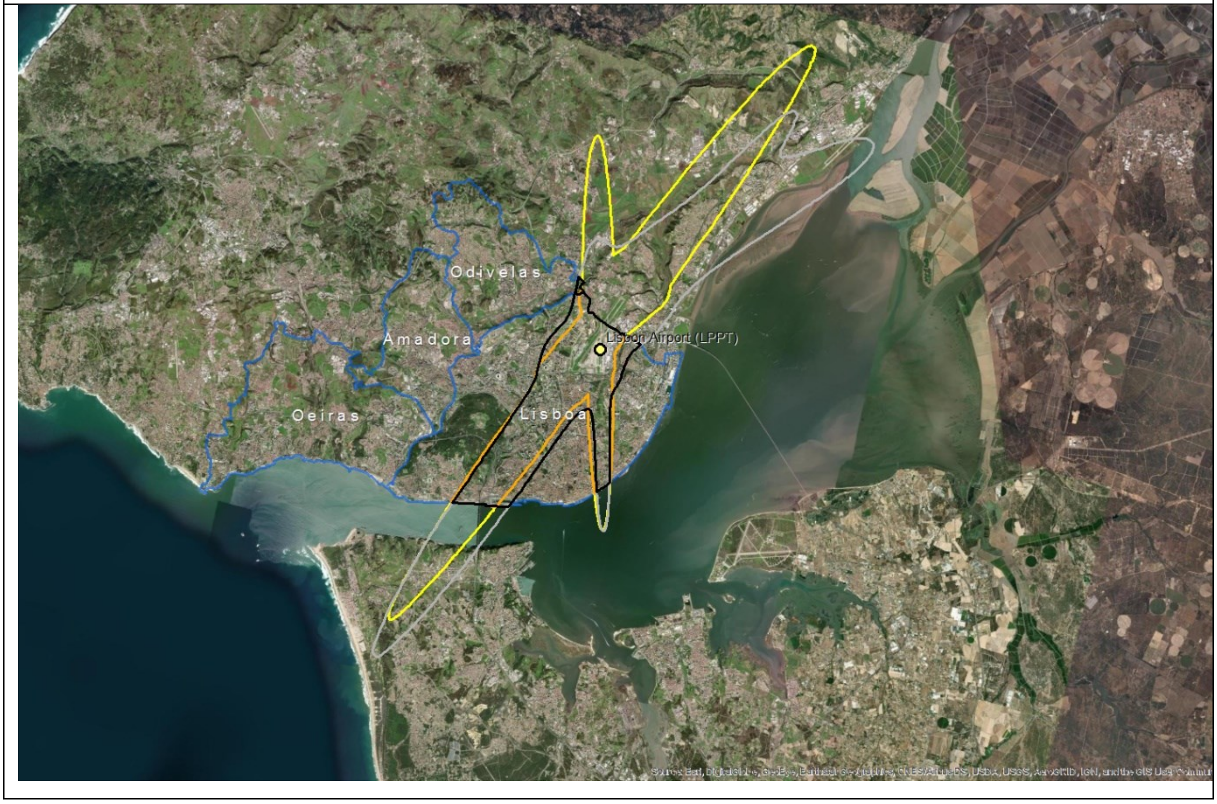
Hamburg (EDDH)


Hamburg (EDDH)
Hamburg
Source: Esri, DigitalGlobe, GeoEye, Earthstar Geographics, CNES/Airbus DS, USDA, USGS, AeroGRID, IGN, and the GIS User Commun

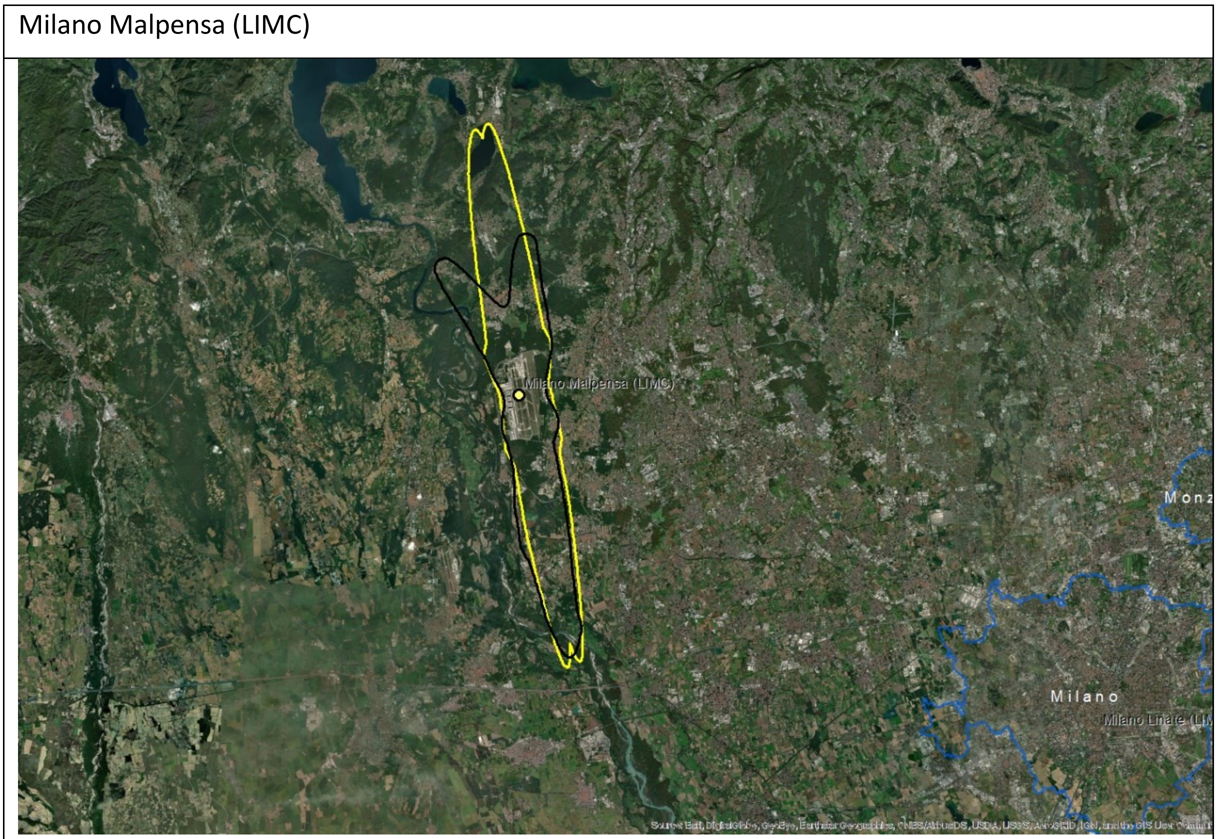
Helsinki – Vantaa (EFHK)


Helsinki-Vantaa (EFHK)
Vantaa
Helsinki-Malmi (EFHF)
Helsinki
Espoo
Kauniainen
Sources: Esri, DigitalGlobe, GeoEye, Earthstar Geographics, CNES/Airbus DS, USDA, USGS, AeroGRID, IGN, and the GIS User Commun

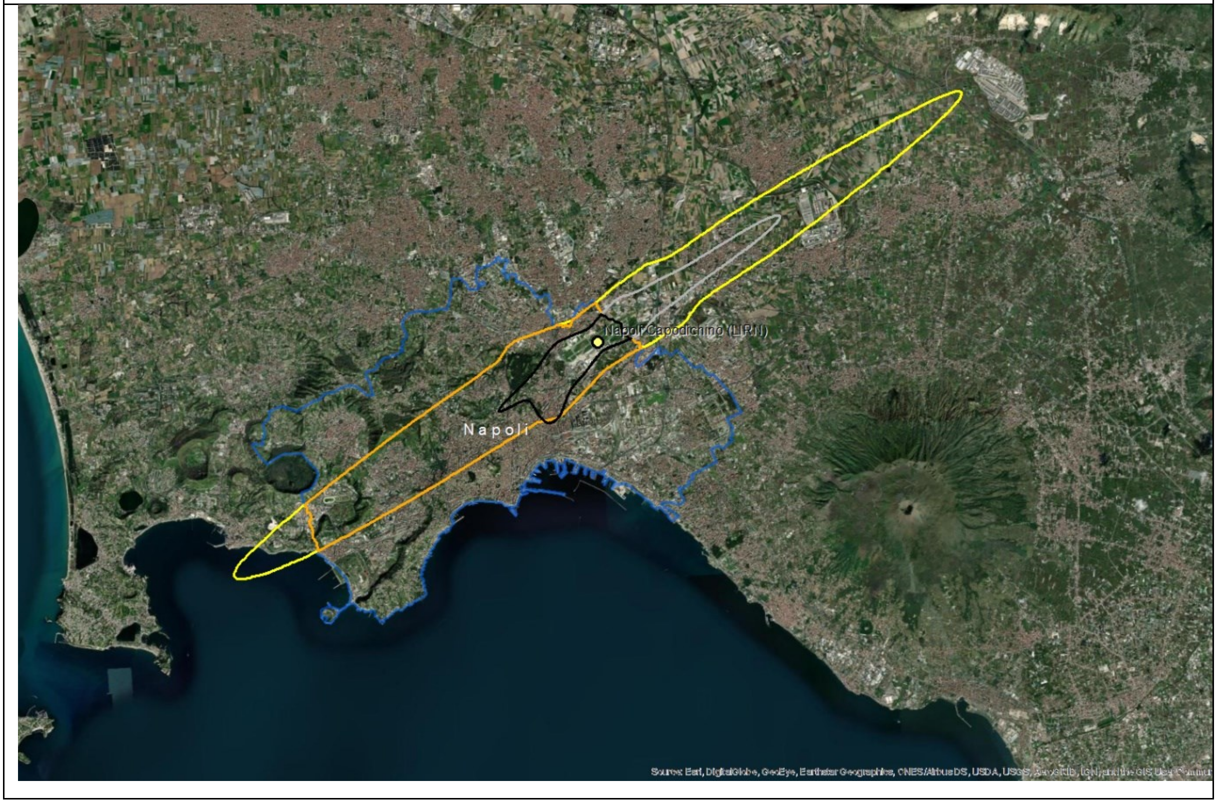
Lisbon (LPPT)


Odivelas
Amadora
Lisbon Airport (LPPT)
Oeiras
Lisboa
Sources: Esri, DigitalGlobe, GeoEye, Earthstar Geographics, CNES/Airbus DS, USDA, USGS, AeroGRID, IGN, and the GIS User Commun

Milano Malpensa (LIMC)


Milano Malpensa (LIMC)
Monz
Milano
Milano Linate (LIM
Source: Esri, DigitalGlobe, GeoEye, Earthstar Geographics, CNES/Airbus DS, USDA, USGS, AeroGRID, IGN, and the GIS User Commun

Napoli (LIRN)


Napoli Capodichino (LIRN)
Napoli
Source: Esri, DigitalGlobe, GeoEye, Earthstar Geographics, CNES/Airbus DS, USDA, USGS, AeroGRID, IGN, and the GIS User Commun

European Topic Centre on Air pollution,
transport, noise and industrial pollution
c/o NILU – Norwegian Institute for Air Research
P.O. Box 100, NO-2027 Kjeller, Norway
Tel.: +47 63 89 80 00
Email: etc.atni@nilu.no
Web : https://www.eionet.europa.eu/etcs/etc-atni

The European Topic Centre on Air pollution, transport, noise and industrial pollution (ETC/ATNI) is a consortium of European institutes under a framework partnership contract to the European Environment Agency.

European Environment Agency
European Topic Centre on Air pollution,
transport, noise and industrial pollution