EERAdata

DATA-DRIVEN DECISION-SUPPORT TO INCREASE ENERGY EFFICIENCY THROUGH RENOVATION IN EUROPEAN BUILDING STOCK

D4.5 – Guidelines for Data Cleaning and Preparation

[WP4 – Data Collection and Integration]

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 847101. The contents of this publication are the sole responsibility of the authors and can in no way be taken to reflect the views of the European Commission.

Lead Contributor	Tobi Elusakin, Trilateral Research
	Tobiloa.elusakin@trilateralresearch.com
Other Contributors	Katrina Petersen, Trilateral Research
	Rafael Abad Cano, COAMalaga
	Karolina Jankowska, ITTI

Due Date	31.07.2021
Delivery Date	29.07.2021
Type	Report
Dissemination level	PU = Public

Keywords	Building stock, data cleaning, inspection, verification, data quality, data collection

Document History

Version	Date	Description
V0.1	06.07.2021	First Draft
V0.2	21.07.2021	After internal review
V1.0	29.07.2021	Final version

Imprint

This document is issued by the consortium formed for the implementation of the EERAdata project under Grant Agreement N° 847101 by the following partners:

TUM- Technische Universität München (Germany)
TRILATERAL – Trilateral Research LTD (United Kingdom)
ITTI – ITTI SP ZOO (Poland)
DTU – Danmarks Tekniske Universitet (Denmark)
KSSENA –Zavod Energetska Agencija Za Savinjsko Salesko in Korosko (Slovenia)
COAMá – Colegio official de Arquitectos de Malaga (Spain)
AEE – Agencia Andaluza de la Energia (Spain)
COP – Kobenhavns Kommune (Denmark)
MOV– Mestna Obcina Velenje (Slovenia)

Disclaimer

Neither TUM nor any other consortium member nor the authors will accept any liability at any time for any kind of damage or loss that might occur to anybody from referring to this document. In addition, neither the European Commission nor the Agencies (or any person acting on their behalf) can be held responsible for the use made of the information provided in this document.

About the project

The EERAdata project will develop and test a decision-support tool to help local administrations in the collection and processing of their building and demographic data towards an assessment and prioritisation of Energy Efficiency measures in planning, renovating and constructing buildings.

While EU policy assigns a primary role to Energy Efficiency (EE), the lack of a holistic understanding of the impact of EE investments has hindered its integration in the policy-making process. Coordination between demand and supply side of energy policy is not targeted, and there is need to gather the evidence on the benefits of EE in ecological and socio-economic terms as well as on its interactions with the broader policy context and energy market.

Project's goals

The project aims to develop:

- Guidelines and roadmaps for the advancement of the clean energy transition
- Joint thematic studies and analyses reports on territorial needs and decarbonisation pathways
- A fully developed and tested decision-support tool to help local administrations in the collection and processing of their building and demographic data towards an assessment and prioritization of EE measures in planning, renovating and constructing buildings

Table of contents

Executive summary

EERAdata is a data-driven project. The approach is built on collecting, preparing, and analysing data. Based on these three activities, we will develop a decision-support tool, along with various other outputs from the project that leverage the data. This deliverable, *D4.5 - Guidelines for Data Cleaning and Preparation* is based on task 4.5 of the EERAdata grant agreement which involves the development and description of models for data collection, cleaning, and preparation. It is part of *WP4 – Data collection and integration* and as such is informed by the activities in tasks 4.1-4.5.

This report presents guidelines on how data can be cleaned and prepared, making it of appropriate quality to be utilised or stored. These guidelines can be used by decision-makers (such as building stock managers or public administrators) in local or regional governments to clean building stock data before it is imported into the DST or used to provide insight into building renovations. This report also presents the data collection and import process for EERAdata, showing a UML notation of the process from downloading the template file from the DST to successfully importing the data back to the DST. A data collection test case with one of the frontrunner partners, Andalusian Energy Agency (AEA) is performed, showing the buildings for which data has been collected and the variables which form the basis of the collected data.

The characteristics of quality data are then presented, with features such as validity, accuracy, completeness, consistency, homogeneity, relevancy, and understandability identified and explained. Following this, a data cleaning process which includes identifying, cleaning and preparation of data is presented and delineated with examples on how data cleaning can be done manually and programmatically.

List of figures

List of tables

1 Introduction

The EERAdata project will conduct research and develop a decision support tool (DST) for calculating the impact of building renovations on energy efficiency (EE). The project aims to gather, sort and inspect various kinds of data formats in order to provide the most detailed information on the building stock of the frontrunner partners: The City of Copenhagen (COP), the Municipality of Velenje (MOV) and the Andalusian Energy Agency (AEA). This deliverable, *D4.5 - Guidelines for Data Cleaning and Preparation* is based on task 4.5 of the EERAdata grant agreement which involves the development and description of models for data collection, cleaning and preparation. It is part of *WP4 – Data collection and integration*. As such, it is informed by D4.1 – D4.4 developed within the same work package.

Data for this project, which includes building-stock, socio-economic and life-cycle assessment data, has been collected from frontrunner municipalities as well as openly available structured and unstructured databases to bolster its quality and completeness. Following the data collection, the data is then prepared and cleaned before it is stored in a number of databases and used to test the DST. Though it may initially present like a linear process, data cleaning and preparation is iterative, meaning it must be performed regularly during the course of data analysis as different issues emerge and more data is gathered. According to Wickham (2014), a dataset consists of values which can be quantitative (numbers) or qualitative (strings), with each value attributable to a variable and an observation. A clean dataset is one in which each variable forms a column, each observation forms a row, and each type of observational unit forms a table. For EERAdata, this means each building property forms a column, each building forms a row, and each municipality forms a table.

Data quality remains a point of concern in relation to buildings and renovations, as dirty data can lead to poor analysis and error-prone decision-making (Chu *et al*., 2016). According to Rahm and Do (2000), data cleaning is the process of identifying and eliminating errors and irregularities from data which has been gathered or collected to enhance its quality. Data cleaning and preparation typically consists of three stages: identification or detection, cleaning, and verification.

The overwhelming majority (about 90%) of the data collected for EERAdata has been inundated with issues such as incomplete data, inconsistent units, irrelevant data, and incorrect values. As a result of the experiences gained in order to develop EERAdata's cleaning methodology, the project has identified a number of guidelines which decision-makers of varying skill levels in different geographical areas can use for data preparation and cleaning.

This report will cover the use of a data template created to aid users in data collection, the characteristics of quality data, and a data cleaning workflow which can be followed in order to attain quality data. The guidelines put forth in this report are based on the experiences preparing and cleaning the data collected for the EERAdata project and will provide guidance to potential users of the tool on how data should be prepared before being fed into the DST or use to develop further insights. The flowchart in figure

1 shows the overarching process for potential users of the DST to collect and prepare their data.

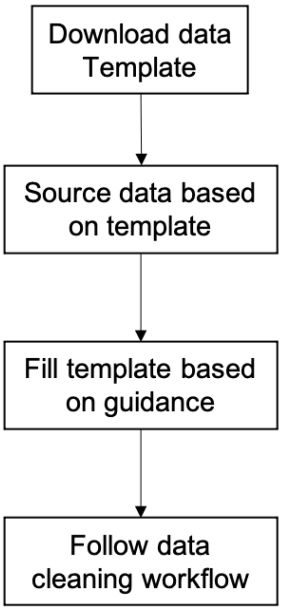

Download data Template
Source data based on template
Fill template based on guidance
Follow data cleaning workflow

Figure 1 Overarching data collection and cleaning process

At this point of the project (M25), the data collection process is still ongoing and an update to the data import process in this deliverable will be provided in the coming months.

The rest of this report is structured as follows: section 2 details the process of data collection within EERA and how it raised novel data cleaning challenges. Section 3 provides guidelines and best practices for data cleaning and preparation developed from engaging those challenges. Section 4 discusses next steps for the project and section 5 concludes the report with insights into data cleaning benefits and challenges unique to energy efficiency and buildings.

2 EERAdata data collection and import process

Data collection is an important part of the EERAdata project with one of its main aims to help local administrations in the collection of their building stock data. A process was therefore developed to aid municipalities and administrators in gathering data, checking its validity, and importing the data to the DST.
This process has been developed based on the calculation modules of the DST (socio-economic, energy demand, indoor environment, supply-side and life-cycle assessment) which are specific to the EERAdata project and aid in the impact assessment of EE-based building renovations. This data collection and import process has been visualised using UML notation in figure 2 and mainly involves the application of a universal data template.

2.1 Data template

The data template containing all the desired variables required for the DST to function has been developed for users of the tool. Potential users of the tool can use the template to identify the variables required for assessment and input the data which corresponds to those variables. A snapshot of a section of the template can be seen in figure 3. In order to download the data template, a registered user will have to go through the following steps:

- Log into the DST with their designated credentials.
- Click on the “Buildings” tab.
- Export the data template to download it to a local device (e.g., a laptop or tablet).

The user will then have to fill the template based on the guidance provided within. This guidance includes information on the name of each variable, a description of what each variable represents, the minimum and maximum thresholds for each variable, the level of importance of each variable to the DST, how each variable can be calculated if not directly available and the unit of each variable. The template has been hardcoded in the following ways to assist municipalities in providing data as accurately as possible:

- Identify values which lie outside the minimum and maximum value thresholds. For example, if the range for building volume is 150m^3 – 45000m^3, then a value of 120m^3 will be highlighted as an error.
- Identify cells where values are missing. If a variable is regarded as required, rows in which values for that variable are missing are also highlighted.
- Identify cells which contain values inconsistent with the variables they have been ascribed to. For example, if a Boolean variable requires True or False values and is given 1’s and 0’s or Yes and No.

2.2 Manual data import

Registered users of the DST who have “expert” access to the tool can also manually input data directly into the DST. This can be done by selecting buildings of interest in the “Buildings” tab within the tool, downloading the properties, inputting the values, and re-uploading the properties. This can be used as an added layer of data quality assurance and error management as issues like data gaps mistakes can be corrected manually for individual buildings. Whole buildings can also be created manually with expert access in the DST with properties and characteristics added for each building. The data provided manually can be used to override default data and data provided in the template. For more information on the user interface for regular and expert users, readers can refer to D3.4 (A report on the finalised EERAdata methodology and tool requirements of the end-users).

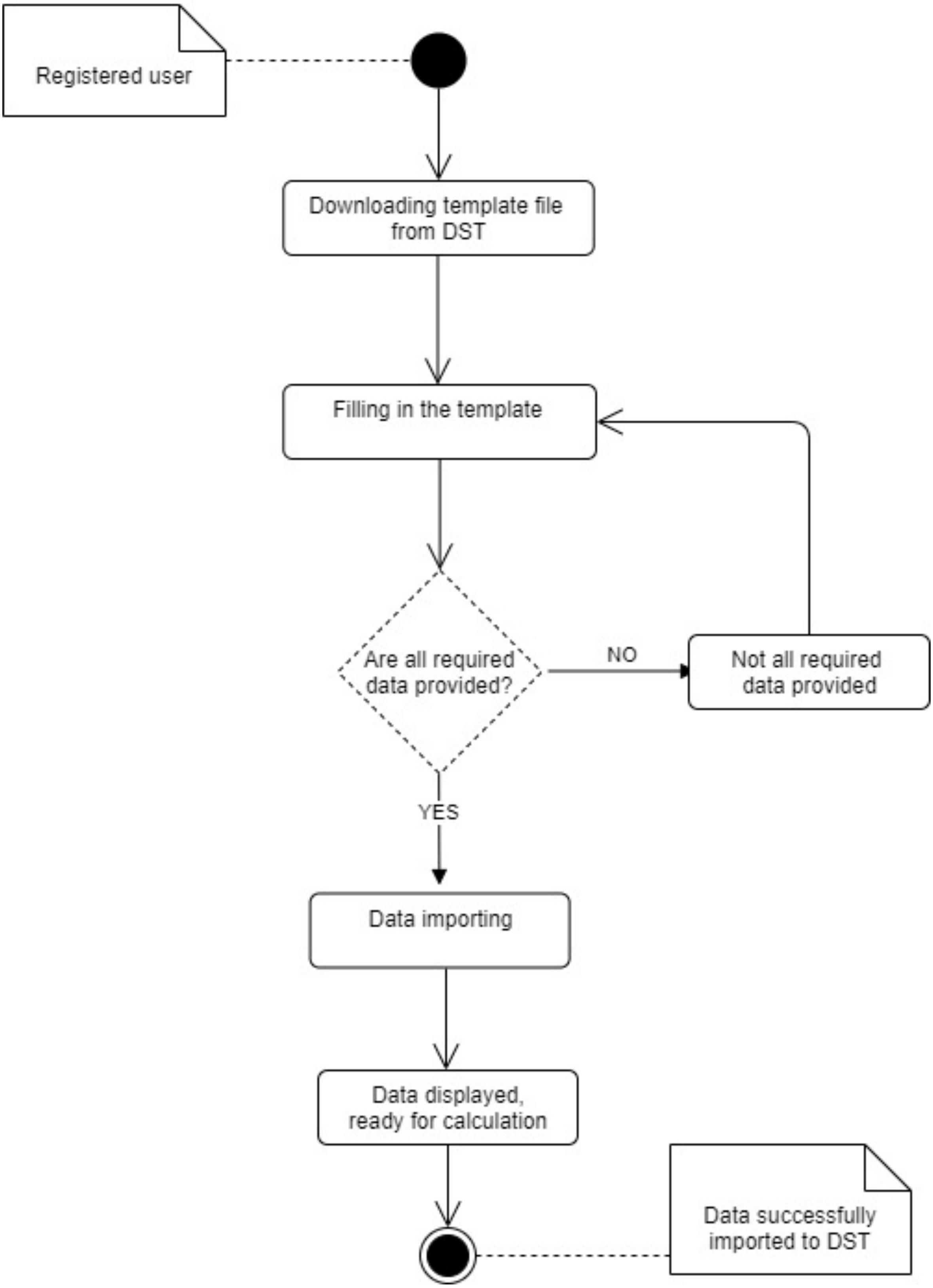
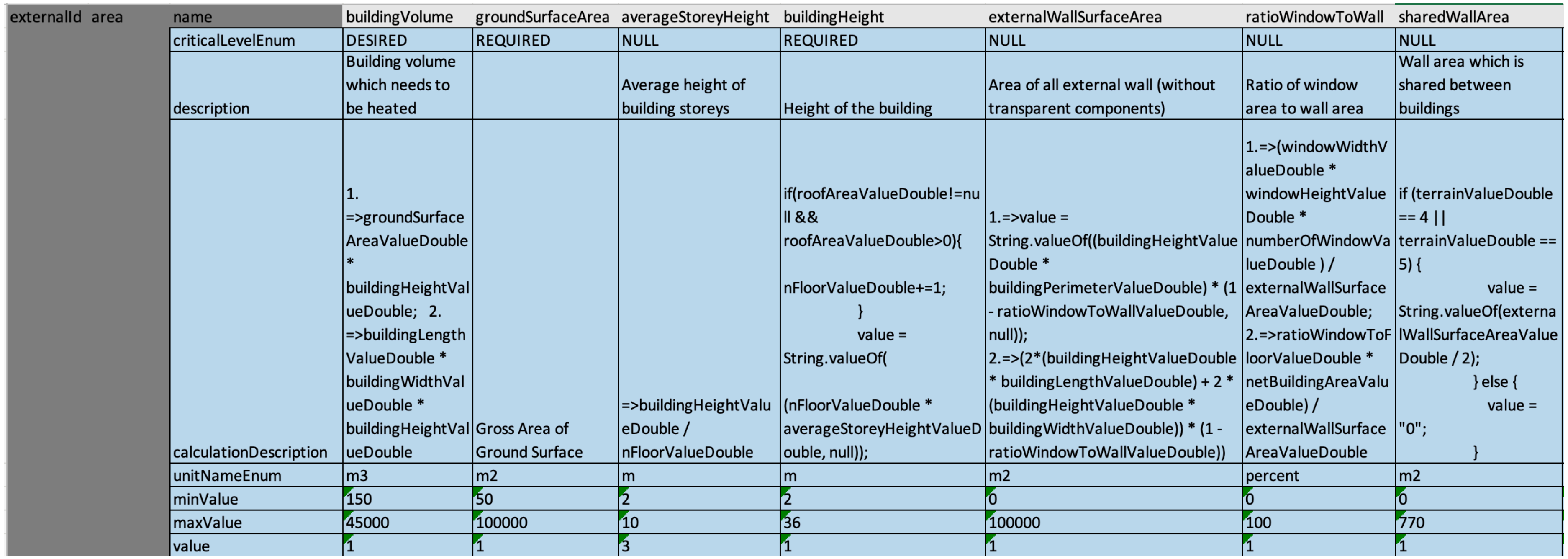

Registered user
Downloading template file from DST
Filling in the template
Are all required data provided?
NO
Not all required data provided
YES
Data importing
Data displayed, ready for calculation
Data successfully imported to DST

Figure 2 UML notation of data import to DST



externalId	area	name	buildingVolume	groundSurfaceArea	averageStoreyHeight	buildingHeight	externalWallSurfaceArea	ratioWindowToWall	sharedWallArea
		criticalLevelEnum	DESIRED	REQUIRED	NULL	REQUIRED	NULL	NULL	NULL
		description	Building volume which needs to be heated		Average height of building storeys	Height of the building	Area of all external wall (without transparent components)	Ratio of window area to wall area	Wall area which is shared between buildings
		calculationDescription	1. =>groundSurfaceAreaValueDouble * buildingHeightValueDouble; 2. =>buildingLengthValueDouble * buildingWidthValueDouble * buildingHeightValueDouble	Gross Area of Ground Surface	=>buildingHeightValueDouble / nFloorValueDouble	if(roofAreaValueDouble!=null && roofAreaValueDouble>0){ nFloorValueDouble+=1; } value = String.valueOf((nFloorValueDouble * averageStoreyHeightValueDouble, null));	1.=>value = String.valueOf((buildingHeightValueDouble * buildingPerimeterValueDouble) * (1 - ratioWindowToWallValueDouble, null)); 2.=>(2*(buildingHeightValueDouble * buildingLengthValueDouble) + 2 * (buildingHeightValueDouble * buildingWidthValueDouble)) * (1 - ratioWindowToWallValueDouble))	1.=>(windowWidthValueDouble * windowHeightValueDouble * numberOfWindowValueDouble) / externalWallSurfaceAreaValueDouble; 2.=>ratioWindowToFloorValueDouble * netBuildingAreaValueDouble) / externalWallSurfaceAreaValueDouble	if (terrainValueDouble == 4 || terrainValueDouble == 5) { value = String.valueOf(externalWallSurfaceAreaValueDouble / 2); } else { value = "0"; }
		unitNameEnum	m3	m2	m	m	m2	percent	m2
		minValue	150	50	2	2	0	0	0
		maxValue	45000	100000	10	36	100000	100	770
		value	1	1	3	1	1	1	1

Figure 3 Snapshot of EERAdata data template

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 847101. The contents of this publication are the sole responsibility of the authors and can in no way be taken to reflect the views of the European Commission.

2.3 Data collection test case - Andalusia

A data template with 54 building-specific parameters was populated by one of the frontrunner partners – Andalusian Energy Agency (AEA). The data import file provided by this partner consists of 11 buildings and incorporates half of the required number of parameters. The buildings studied for data import include:

1. Public administrative office building. Located in Seville. The energy certificate of this building has been obtained with CE3X, the Spanish simplified official tool for public buildings, medium and large tertiary.
2. The Andalusian Energy Agency building. Located in Seville. The energy certificate of this building has been obtained with CE3X, the Spanish simplified official tool for public buildings, medium and large tertiary.
3. Heliopolis administrative elderly people center. Located in Seville. The energy certificate of this building has been obtained with CE3X, the Spanish simplified official tool for public buildings, medium and large tertiary.
4. Andalusian Health Service. Located in Seville. The energy certificate of this building has been obtained with CE3X, the Spanish simplified official tool for public buildings, medium and large tertiary.
5. Antequera Hospital. Located in Malaga. The energy certificate of this building has been obtained with CE3X, the Spanish simplified official tool for public buildings, medium and large tertiary.
6. Modern section of Environmental Ministry building, known as “Casa Sudhein”. Located in Seville. The energy certificate of this building has been obtained with CE3X, the Spanish simplified official tool for public buildings, medium and large tertiary.
7. Original section of Environmental Ministry building. Located in Seville. The energy certificate of this building has been obtained with CE3X, the Spanish simplified official tool for public buildings, medium and large tertiary.
8. The government office. Located in Malaga. The energy certificate of this building has been obtained with CE3X, the Spanish simplified official tool for public buildings, medium and large tertiary.
9. San Telmo Palace. Located in Seville. The energy certificate of this building has been obtained with CE3X, the Spanish simplified official tool for public buildings, medium and large tertiary.
10. Johannes Kepler Building. Located in Seville. The energy certificate of this building has been obtained with CE3X, the Spanish simplified official tool for public buildings, medium and large tertiary.
11. Social housing building. Located in Malaga. The energy certificate of this building has been obtained with HULC, the Spanish unified official tool for housing, medium and large tertiary.

A variety of sources were mined in a joint effort by COAMalaga and A.E.A to fill in the data template for Andalusia. These include energy certificates, cadastres, certification tools, expert assessment, and other parameters (arithmetic calculation from known parameters). The two official certification tools which provided most of the data included:

- HULC – created by merging two previous official tools (Lider and Calener). This is the tool used in the files received from AVRA (the Social Housing Agency from Andalusia).
  CE3X – Simplified tool, dedicated to public buildings. This is the tool used by the AEA (Andalusia Energy Agency).

The data mined from the aforementioned sources were fed into the DST template, cleaned by Trilateral and imported into the DST through its file import function.

Table 1 Summary of sources for each parameter

Num	Parameter	Energy Certificate	HULC file	CE3x file	Cadastre	Expert asses.	Other param.
1	netBuildingArea	X	X	X			
2	name	X	X	X			
3	buildingVolume	X	X	X			
4	groundSurfaceArea		X	X	X	X	
5	averageStoreyHeight	X	X	X			
6	buildingHeight		X				X
7	buildingperimeter				X	X	
8	externalWallSurfaceArea	X	X	X		X	
9	ratioWindowToWall					X	X
10	ratioWindowToFloor					X	X
11	gValueWindows		X	X		X	X
12	frametypeofwindow		X	X		X	X
13	Glazingofwindow		X	X			
14	sharedWallArea		X	X			
15	uValueExternalWall	X	X	X			
16	uValueWindows	X	X	X		X	X
17	uValueBasePlate	X	X	X			
18	uValueRoof	X	X	X			
19	uValueSharedWall	X	X	X			
20	roofArea	X	X	X			
21	heatedTopFloor		X	X		X	
22	percentageBoilerGasHeating	X	X	X			
23	percentageBoilerOilHeating	X	X	X			
24	percentageHeatPumpsHeatingAirWater	X	X	X			
25	percentageHeatPumpsHeatingWaterWater	X	X	X			
26	percentageHeatPumpsHeatingSoleWaterCollector	X	X	X			
27	percentageHeatPumpsHeatingSoleWaterSonde	X	X	X			
28	percentageElectricHeating	X	X	X			
29	percentageDistrictHeatingHeatingNonRenewable	X	X	X			
30	percentageBiomassHeating	X	X	X			
31	percentageSolarSystems	X	X	X			
32	percentageDistrictHeatingRenewable	X	X	X			
33	percentageEFlowHeater	X	X	X			
34	percentageGasStorage	X	X	X			
35	yearOfConstruction	X	X	X	X		

36	percentagePipeSystemSurfaceHeatingHeatingOriginal	X	X	X			
37	percentagePipeSystemRadiatorsHeatingOriginal	X	X	X			
38	nFloor		X	X			
39	buildingWidth				X	X	X
40	buildingLength				X	X	X
41	thermalMass					X	
42	normalizedLeakageArea					X	
43	windowWidth		X	X		X	
44	windowHeight		X	X		X	
45	openingAreaRatio					X	
46	numberOfWindows		X	X		X	
47	frameWindowRatio		X	X		X	
48	solarShading					X	
49	otherHeat					X	
50	materialSensitivity					X	
51	frameArea	X	X	X		X	
52	frameU	X	X	X		X	
53	glazingU	X	X	X		X	
54	heatExchangeEff		X	X			

3 Guidelines for Data Cleaning and Preparation

According to Dasu and Johnson (2003), data cleaning and preparation can take up 80% of the entire data analysis process and is performed in order to attain quality data. Issues with data quality are prevalent in both single data collections and large database systems but require similar actions to overcome. This section presents a broad definition of data quality and provides guidelines and best practices on how quality can be assured through data cleaning and preparation. The data cleaning process outlined in this section, with its diagrammatic workflow and code-based examples, has been uniquely developed to be aid the impact assessment of energy efficiency-based renovations on the energy efficiency of buildings.

3.1. Data quality

Data quality can best be described as “fitness for purpose”, meaning the measure of quality of any piece of data depends on what it is to be used for. This makes data quality a relative and multidimensional concept as organisations and individuals must deal with data based on its intended use, the subjective perception of individuals tasked with analysis and the objective analysis of the data in question (Tayi and Ballou, 1998; Pipino *et al*., 2002). In relation to EERAdata, quality data refers to data which can be used to assess the impacts of building renovations on energy efficiency. This includes data based on the socio-economic, indoor climate, life-cycle assessment, energy demands and supply-side assessment methodologies.

For data of any kind to be deemed as quality, there are a number of conditions it must fulfil. These include data validity, accuracy, completeness, consistency, homogeneity, relevancy, and comprehensibility (Kahn *et al*., 2002). These features are elaborated on below:

- **Validity**: The validity of any dataset refers to the extent to which the data fits with pre-defined organisational procedures and constraints. These constraints can be based on the data type (e.g., integer, float, Boolean, date-time, etc.), range of data (i.e., dates should fall within a certain range), foreign key (as seen in relational databases), or categories (such as male and female to describe gender). An example of validity in EERAdata can be seen in the variable "net building area". If the data type presents as Boolean instead of float, the data is deemed invalid.
- **Accuracy**: The accuracy of a dataset indicates the closeness of the data to the true values. Values in a dataset can be deemed as valid, but not accurate. Using the EERAdata building stock database as an example, a building address can be valid, meaning it meets all of the pre-defined constraints, but inaccurate in that it might not actually exist.
- **Completeness**: The completeness of a dataset refers to the degree to which data is known and is not missing. Data can be missing for several reasons including lack of measurement in the first place, difficulty in assessing the data source and data entry errors. This issue can be mitigated by actually measuring the data required, re-evaluating or questioning the data source, and using averages and other calculations for numerical values.
- **Consistency**: The consistency of data is defined as the extent to which data is presented in the same format within and across datasets. For example, a valid building height of 12m does not match well with a number of floors of 10.
- **Homogeneity**: The homogeneity of data refers to the degree to which the values of a specific variable have the same unit of measure. This feature mostly applies to quantitative (numerical) rather than qualitative (categorical) data and requires that the unit of measure for a variable be pre-defined based on what is required for analysis or visualisation. For example, data representing the net surface area of a building should have one unit of measure (i.e., mm^2 or m^2 or km^2).
- **Relevancy**: This refers to the degree to which data is relevant to the task or project in question. This means that data can be valid, accurate, complete, consistent and uniform but will prove useless if it cannot be applied to the task at hand.
- **Comprehensibility**: Data comprehensibility refers to the extent to which data can be easily understood. This means that data collected for any task or project must possess features that enable it to be read and understood easily by its users. This includes using suitable language, symbols and units of measurement.

3.2. Data cleaning process

The data cleaning process involves all actions taken to ensure the data collected embody the characteristics of quality data. The workflow can be distilled into three broad phases: inspection, which involves the detection of issues/errors in the data; cleaning, which involves addressing the issues identified; and verification which involves confirmation that the results of the cleaning process are correct. Figure 4 shows a flowchart depicting the data cleaning workflow.

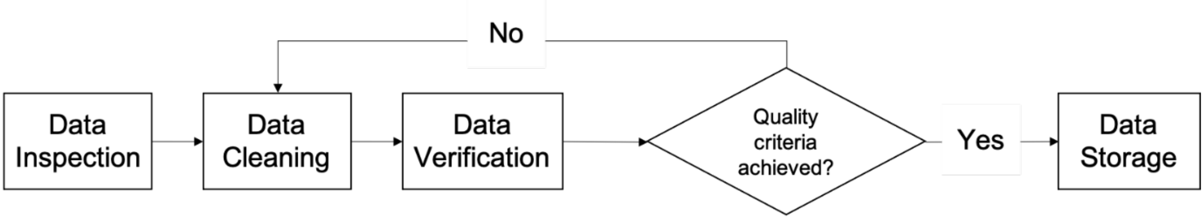
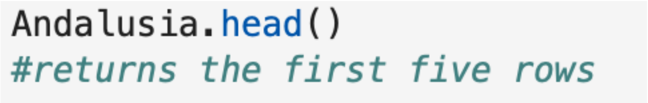


Figure 4 Data cleaning workflow

3.2.1. Inspection

Data inspection refers to the process of identifying/detecting errors affecting the quality of the data. Given that data is mainly collected in csv or xlxs format, inspection can be done manually (humans looking through the data sheet) or programmatically (using code). Pandas, a data analysis library in the Python programming language, is a good tool for inspecting and detecting errors in data tables. For data cleaning processes which involve using individual buildings as observations, inspection helps to identify areas where data for different buildings may be conflated or where data for certain buildings are unavailable but can be calculated. This process can be time consuming and involves a number of methods:

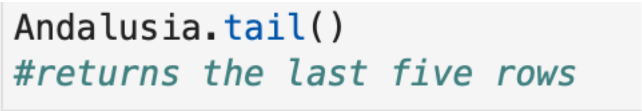
- **Inspecting table head and tail**: This action can be performed after the data table has been read into the Pandas library as a dataframe. A dataframe is a two-dimensional marked data configuration with potentially different column types[1]. The head method produces the first five rows of the data and the tail method produces the last five rows of data. This helps to provide a quick glance at the data table to identify errors which require a quick fix such as data gaps or incorrect column headers. Figures 5, 6 and 7 shows an example Python code for obtaining the first five, last five and first seven rows of data respectively.



```
Andalusia.head()
#returns the first five rows
```

Figure 5 Example code for returning the first five rows

The tail method works the same way:

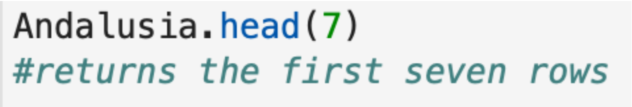


```
Andalusia.tail()
#returns the last five rows
```

Figure 6 Example code for returning the last five rows

The number of rows to be returned can also be specified:

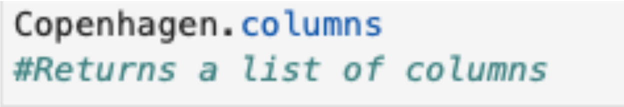
[1] Pandas User Guide – Intro to data structures (https://pandas.pydata.org/pandas-docs/stable/user_guide/dsintro.html)



```
Andalusia.head(7)
#returns the first seven rows
```

Figure 7 Example code for returning the first seven rows

- **Inspecting the column names**: The column names can be inspected by manually looking at all the columns (if the dataframe is small enough) or inspected programmatically using the Pandas method "*.column*". This can be used for situations which require knowledge of the available data points and to determine if there are any spelling mistakes. Figure 8 shows example code for inspecting column names.



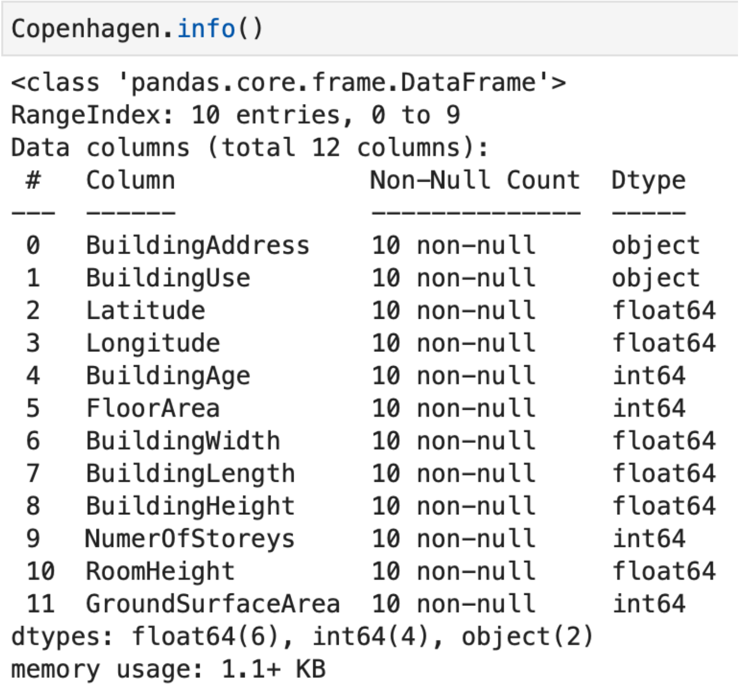
```
Copenhagen.columns
#Returns a list of columns
```

Figure 8 Example code for returning a list of columns

- **Inspecting the data type**: The data type for each column can be obtained and inspected using the Pandas method "*.dtypes*". Different data types in Pandas include:
    - Object (text or a combination of numeric and non-numeric inputs),
    - Int64 (integers),
    - Float64 (numbers with decimal points),
    - Bool (true/false values),
    - Datetime64 (time and date values) and
    - Category (a set list of text values).

If the data type for a column is wrong, e.g., an integer column designated as float or vice-versa, this can be detected using the "*.dtypes*" method and changed.

- **Inspecting for missing values**: Missing values in a Pandas dataframe can be identified either manually or programmatically using the method "*.info()*". When successfully run, this method produces the list of columns, the data type of each column and how many non-null values are present in each column. The number of non-null values signify how many rows contain actual values. The "*.info()*" method is versatile in that it can also be used to inspect column names as well as data types. Figure 9 shows example code for inspecting missing values.

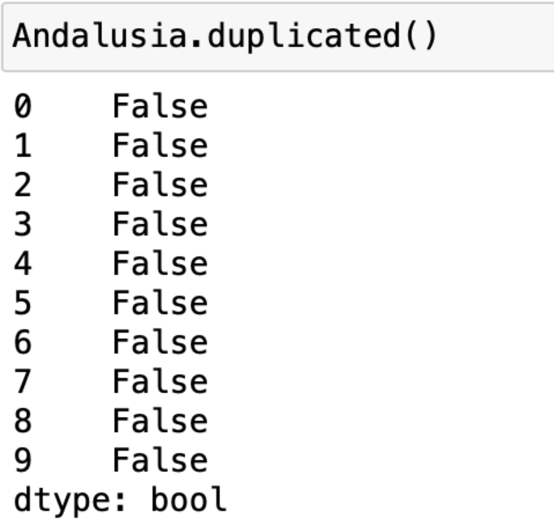


```
Copenhagen.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 12 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   BuildingAddress     10 non-null     object
 1   BuildingUse         10 non-null     object
 2   Latitude            10 non-null     float64
 3   Longitude           10 non-null     float64
 4   BuildingAge         10 non-null     int64
 5   FloorArea           10 non-null     int64
 6   BuildingWidth       10 non-null     float64
 7   BuildingLength      10 non-null     float64
 8   BuildingHeight      10 non-null     float64
 9   NumerOfStoreys      10 non-null     int64
 10  RoomHeight          10 non-null     float64
 11  GroundSurfaceArea   10 non-null     int64
dtypes: float64(6), int64(4), object(2)
memory usage: 1.1+ KB
```

Figure 9 Example of .info() method in Python

- **Inspecting for duplicate data**: Duplicate data can also be identified manually (through visual inspection) or through code. The latter can be done using the Pandas method "*.duplicated()*". The duplicated method provides a Boolean series with the value "True" showing duplication, i.e., the entry is identical to a previous one. Figure 10 shows an example of this method.



```
Andalusia.duplicated()
```

```
0    False
1    False
2    False
3    False
4    False
5    False
6    False
7    False
8    False
9    False
dtype: bool
```

Figure 10 Example of .duplicated() method in Python

3.2.2. Cleaning

The data cleaning workflow involves a several processes which can be applied based on the type, size and format of the data collected. Data cleaning helps to eliminate issues with the data collected which might contribute to faulty analyses by the DST, therefore leading to the wrong decision being made regarding energy efficiency-based renovations. The Pandas library in Python was found to be best practice for data cleaning with the following processes:

- **Converting data types**: This can be done when faced with the problem of incorrect data types and is best done programmatically using Pandas. For example, if the column "NumberOfStoreys" in figure 7 presents as a floating-point number (float64) rather than an integer (int64). This problem can be corrected using the "*.astype*" method.
- **Handling unnecessary/duplicated data**: Unnecessary data refers to data which is not needed or does not fit the context of the project in question. Duplicated data can occur in both rows and columns. This can be addressed manually by simply deleting the unwanted rows or columns if the dataset is not too large or by using the "*.drop*" function in Pandas to remove unnecessary or unwanted data by specifying the column name or row number. For example, phone numbers are not needed for analysing the building stock of a municipality or city. Hence, if this is a column in the dataset, it should be dropped. In a similar vein, if we only need non-residential buildings for analysis, any observations (rows of data) which contain residential buildings can be removed.
- **Filling missing data**: Missing data or data gaps is one of the most common issues in a dataset. There are a number of ways to deal with gaps in data:
    - **Input data**: This refers to the process of filling gaps in a dataset by calculating based on other values. Statistical values such as the mean and median can be used but do not guarantee impartial data. Missing data can also be calculated using values from other variables. For example, the volume of a building can be calculated if the net surface area and building height are known. Linear regression can also be applied to available data to determine the best fit line between two variables.
    - **Drop data**: This is the process of dropping rows or columns of data which contain missing values. If very few of a column's values are missing and this occurs at random, it is recommended that the rows which contain the missing values be dropped. If the majority of a column's values are missing, it is recommended that the whole column be dropped instead.
- **Addressing inconsistent values**: This is the process of ensuring that all observations in a column are consistent in their format. For categorical or qualitative data, this can be done using the "*.replace*" function in the Pandas library to replace the wrong values with the correct ones. For numerical values, there are different ways inconsistent values can be corrected. An example is using the "*.round()*" function to set numerical values to the preferred number of decimal points.
- **Renaming Columns**: This involves selecting column names with typos and errors as well as non-descriptive column names and replacing them with full

words absent of typos. In the Pandas library this can be done using the "*.rename()*" function.

3.2.3. Verification

This is the process of reviewing and re-inspecting the data to make sure the steps taken to clean the data and the pre-defined constraints are met. The verification process is important and can be done using verification scripts in Python or conducting a visual inspection of the data manually. If the verification process reveals gaps or errors, the data cleaning workflow should be repeated, and this can be done iteratively until the data is fit for purpose.

4 Conclusion

In conclusion, data cleaning and preparation are fundamental to ensuring the quality of data for any project. This is made even more imperative when dealing with building stock data which can be used to make decisions on and assess the impact of investments in energy efficiency-based renovations. To that end, it is important that potential users of the tool are aware of the best practices which can be used to guarantee quality data for import into the DST and for alternative use.

This deliverable has therefore identified and outlined several characteristics of quality data, following which a number of guidelines and best practices on how data should be cleaned were proposed. The guidelines, which are sectioned into inspection, cleaning, and verification, were identified based on the data cleaning work performed in EERAdata. Following the cleaning process, data can be imported to the DST for use in impact assessment or can be stored as an exploitable result to be analysed for insight by decision/policy makers in public administrations. It is imperative for data to be as clean as possible in order that insight obtained from the analysis can be as useful as possible. Data analysis for EERAdata provides insight into the building stock of the respective frontrunner municipalities, helping to get a good understanding of the building, socio-economic, indoor environment, and energy demand characteristics.

It is also important to identify and address challenges to data utilisation and analysis as early as possible in any project, in order to avoid the "garbage in, garbage out" situation in which data which is not cleaned properly or at all leads to faulty insights and poor decision-making. This document equips all individuals/organisations working with building stock/energy efficiency data with the understanding and tools to put the data to good use.

This deliverable has also provided information on the process of importing data into the DST with a UML notation diagram used to visualise the process.

References

Dasu, Tamraparni, and Theodore Johnson. *Exploratory data mining and data cleaning*. Vol. 479. John Wiley & Sons, 2003.

Tayi, Giri Kumar, and Donald P. Ballou. "Examining data quality." *Communications of the ACM* 41.2 (1998): 54-57.

Pipino, Leo L., Yang W. Lee, and Richard Y. Wang. "Data quality assessment." *Communications of the ACM* 45.4 (2002): 211-218.

Kahn, Beverly K., Diane M. Strong, and Richard Y. Wang. "Information quality benchmarks: product and service performance." *Communications of the ACM* 45.4 (2002): 184-192.