Table of Contents

[Data selection and cleaning 2](#_Toc58785285)

[Identification of House price data: 2](#_Toc58785286)

[Obtaining the House price dataset: 2](#_Toc58785287)

[Cleaning of the house price dataset: 2](#_Toc58785288)

[Identification of broadband data: 2](#_Toc58785289)

[Obtaining the Broadband dataset: 3](#_Toc58785290)

[Cleaning of the broadband dataset: 3](#_Toc58785291)

[Legal and/or ethical issues with the proposed system: 3](#_Toc58785292)

[Structured and semi-structured data 4](#_Toc58785293)

[Data model and implementation 5](#_Toc58785294)

[SQL 5](#_Toc58785295)

[R 6](#_Toc58785296)

# Data selection and cleaning

## Identification of House price data:

The data set related to house prices is taken from the website Data from Office of National Statistics (ONS): “Median price paid by ward, England and Wales, year ending Dec 1995 to year ending Dec 2019” and link to gather the data is [https://www.ons.gov.uk/peoplepopulationandcommunity/housing/datasets/medianpricep aidbywardhpssadataset37](https://www.ons.gov.uk/peoplepopulationandcommunity/housing/datasets/medianpricep%20aidbywardhpssadataset37)

## Obtaining the House price dataset:

These data sets having the features related to house prices in UK Government and Broadband speeds for different countries, constituencies as well as ward related data.

Inside the House price dataset, there are many sheets are there for different purposes. From those sheets we are using Median price paid by county, year ending Dec 1995 to year ending Jun 2020 table for our project as it suit most for the questions to be queried with R. The features in this sheet are Region/Country code , Region/Country name, County/UA code, County/UA name, Year ending 1995,… Year ending Jun 2018,Year ending Sep 2018,Year ending Dec 2018,Year ending Mar 2019,Year ending Jun 2019,Year ending Sep 2019,Year ending Dec 2019,Year ending Mar 2020,Year ending Jun 2020 Quarter wise.

## Cleaning of the house price dataset:

To clean the dataset, firstly we have checked for the information available in the dataset. Except first few variables are character type data and remaining all are numeric type data. We checked for any existence of null values present in each column, but there is no null values present in our dataset and all values are null free. Also Before uploading the data into the database first we renamed the all column names like including underscore between the spaces present in the column name. For Ex Country code will be written as Country\_code, Year ending Sep 2019 is written as Year\_ending\_Sep\_2019. Also, before uploading data into the database we need to ensure that whether all data is present in third normal form is or not. A database is in third normal form when it is in second normal form, no non key attribute is transitively dependent on the primary key and all attributes are single valued. Here in our table no single column is multi valued and each value is different from each other.

## Identification of broadband data:

The data set related to Broadband is taken from the website Broadband data file – constituency and ward data UK 2017 <http://researchbriefings.files.parliament.uk/documents/CBP-8200/CBP-8200.xlsx>

## Obtaining the Broadband dataset:

In broadband Dataset, there are three sheets present and we are interested to find out the insights ward wise and we chosen the ward data sheet to get insights and this sheet suitable for the questions in the description.The features inside this column are Constituency,Region/Country,Average Download Speed Mb/s,SFBB or UFBB Availability (>30 Mb/s) %,Fibre to the property availability %,Connections Unable to receive 10 Mb/s %,Connections Receiving Under 2 Mb/s %,Connections Receiving Under 10 Mb/s %,"Connections Receiving Superfast Speeds (>30 Mb/s) %",Connections Receiving Over 300 Mb/s %,Average speed for basic broadband lines (Mb/s),Average speed for superfast lines (Mb/s),Estimated data completeness.

## Cleaning of the broadband dataset:

To clean the broad band dataset, firstly we have checked for the information available in the dataset. Except first few variables are character type data and remaining all are numeric type data. We checked for any existence of null values present in each column, but there is no null values present in our dataset and all values are null free. Also Before uploading the data into the database first we renamed the all column names like including underscore between the spaces present in the column name. For Ex Connections Receiving Superfast Speeds (>30 Mb/s) will be written as Connections\_Receiving\_Superfast\_Speeds\_(>30 Mb/s) and , Average speed for superfast lines (Mb/s) written as Average\_speed\_for\_superfast\_lines\_(Mb/s). Also, before uploading data into the database we need to ensure that whether all data is present in third normal form is or not. A database is in third normal form when it is in second normal form, no non key attribute is transitively dependent on the primary key and all attributes are single valued. Here in our table no single column is multi valued and each value is different from each other.

# Legal and/or ethical issues with the proposed system:

The writers have dedicated all of the programming and documents in SQLite to the public domain. Both code writers and members of the organisations with whom they operate have signed affidavits dedicating their contributions to the public domain, and the originals of such signed affidavits are kept at Hwaci's headquarters in a fire secure. All is free to copy, change, print, use, compile, sell or distribute the original SQLite code for any purpose, commercial or non-commercial, and by any means, either in source code form or as a compiled binary. The previous paragraph refers to the SQLite deliverable code and documents - those pieces of the SQLite library you are currently bundling and delivering for a broader program. Some scripts used as part of the build process (such as the autoconf-generated 'configure' scripts) might come under other open-source licenses. However, nothing ever hits the final deliverable SQLite library from these build scripts, and so licenses aligned with such scripts should not be a factor in deciding the copy and usage privileges for the SQLite library. In SQLite, all of the deliverable functionality has been written from scratch. No other projects or the free Internet have taken some code. Each line of code can be traced back to its original source, and there are public domain dedications on file for both of those writers. Thus the SQLite code base is safe and the approved code from other projects is uncontaminated. SQLite is an open source, meaning that with these copies you can make as many copies as you want and do anything you want. But open-contribution is not SQLite. The project does not allow patch from unauthorized parties in order to protect SQLites in the public domain to ensure that the code may not be infected with proprietary or licensed content. The entire SQLite code, particularly written for SQLite use is original. No code from obscure internet origins has been copied. The project does not allow updates, so that SQLite is entirely free and proprietary. If you like a proposed modification and have a fix as a proof-of-concept, it's perfect. If we rewrite your patch from scratch, please do not be offended.

Similar to the SQLite , RStudio is also completely free to use for an individual and for organization purpose. Although our copyright is freely shared by AGPL version 3, our marks are the exclusive property of RStudio, as expressly allowed under GNU Affero version 3 under Section 7 General Public License (e). There was a mistake. The logos of RStudio serve as a source indication.

The datasets we have taken from publicly available sources and there will be no legal or ethical issues for this.

We can create sqlite databases in many ways, via command line, SQLite studio and SQlite(Browser). Here we are going to create new database via SQLite(Browser) as it will be good in understanding and easy for creating tables.

# Structured and semi-structured data

Structured information is data that is complied with and thus easy to interpret in a predefined data model. Structured data is consistent with a table of connections in the various rows and columns. Excel or SQL databases are typical examples of organized data. Each row and column has been structured and can be sorted.

The structured data relies on the nature of the data model – a model for storing, manipulating and accessing information. Per field is discreet due to a data model and can be accessed either independently or in combination with other fields. This is very powerful for structured data: the data from multiple storage locations can be aggregated easily.

After earliest implementations of database management systems (DBMS) were able to store, process and view structured data, structured data is regarded as the most "traditional" type of data savings.

Semi-structured data is a type of structured data that is non-compatible with the formal framework of relational database modeling or other types of tables but still includes tags or other markers that distinguish semi-semantic elements and introduce hierarchies within the data. It is also known as the structure of self-description. For starters, JSON and XML types of semi-structured data are semi-structured data.

It persists because semi-structured data is slightly simpler to interpret than unstructured data because the third group exists (between structured and unstructured data). Many big data solutions and software will "read" and process JSON or XML. In contrast to unstructured data it reduces the difficulty of the processing of structured content.

There are many advantages of using SQLite over XML and those are

* in one file, all data related to tables can be stored
* The reduction of efficiency of cache is smaller than XML,
* You should hold metadata from the cache itself, but it can be accessed in the same manner,
* SQL is definitely easier to deal with than XPath for most users.

Our data is completely structured and labeled, it will be better to use SQLite for our data. So will suggest going for SQLite instead of Semi structured model(XML).

# Data model and implementation

## SQL

* A table should be in third normal form if it is already in second normal form and there is no transitive dependency exists on primary key and all attributes are single valued.
* In our data, there is no multi valued data existing for each label and there is no transitive dependency exists between any variable.
* Both datasets is normalized into third normal form and we can use these datasets for our analysis.
* In our database, we are having two tables. One table is house data, and we are having two different types of data present in this table. One is numeric and another is char data.
* Similarly, in the broadband table also we are having two different types of data present in this table. One is numeric and another is char data
* There is no relationship existing between two tables and two tables are independent with each other, all values are defined as not null and there is no primary and foreign keys are there in each table.
* After loading the data into data base as tables, we exported the database which is created with two tables.

## R

The main steps of how R code is to be run (executed):

The bridge to access database in R studio is RSQLite package. With help of this library we are going to access the information present inside the tables. The command used to provide connection is dbConnect(RSQLite::SQlite) and we need to mention the database name which is going to be loaded into RStudio.

The libraries required for our analysis is dplyr and RSQLIte. The results we got from the output are,

* For city of London the average prices of houses in a particular year such as 2018 is 886069.5
* For city of London the average prices of houses in a particular year such as 2018 is 906851
* For city of London, the average increase in prices (in percent) between two years ie., 2017 and 2018 is 47332
* Considering all districts (Yorkshire and the Humber), The ward which has the highest house price in Mar 2019 is Harrogate with house price of 274950
* Considering all districts (Yorkshire and the Humber), The ward which has the highest house price in Dec 2019 is Kingston upon Hull, City of with price of 115000
* Based on the data source (you use), broadband speed (average download), in a Alibon ward (or a postcode) of a district is 57.1
* Based on the data source (you use), (superfast) broadband availability (%), in a Alibon ward (or a postcode) of a district is 0.988 MBPS
* The ward is having highest average broadband speed is Upper Lune Valley with 472.8Mbps
* The Constituency is having highest average broadband speed is Morecambe and Lunesdale with 472.8Mbps
* The Region/Country is having highest average broadband speed is North West with 472.8Mbps