Technical (setup) & user guide for the SHINE decision support tool.

Table of Contents

[Pre-amble 3](#_Toc131400268)

[What is the SHINE tool and what does it do? 3](#_Toc131400269)

[Why was SHINE built? 3](#_Toc131400270)

[Clarifying terms 3](#_Toc131400271)

[Referencing / Attribution / Branding: 3](#_Toc131400272)

[License for use: 3](#_Toc131400273)

[Acknowledgements 3](#_Toc131400274)

[For development Q&A and trouble-shooting 3](#_Toc131400275)

[Data sources: 4](#_Toc131400276)

[Branding development 5](#_Toc131400277)

[How to get in touch with feedback / suggestions: 5](#_Toc131400278)

[Section 1: Technical Set-Up 6](#_Toc131400279)

[Setting up the environment from the yaml file 6](#_Toc131400280)

[Folder Structure Required: 6](#_Toc131400281)

[API Key Requirement 8](#_Toc131400282)

[Section 2: User Guide 9](#_Toc131400283)

[user\_and\_data\_params.xlsx File 9](#_Toc131400284)

[Overview 9](#_Toc131400285)

[What to complete in the file? 9](#_Toc131400286)

[Structure 9](#_Toc131400287)

[Explanation of the separate files: 17](#_Toc131400288)

[Explanation of what the user must do when running the program: 19](#_Toc131400289)

[Stage\_01 to stage\_06 (HEA, ML DNA, Emissions): 19](#_Toc131400290)

[Section 3: Explanation of methods and known trade-offs 20](#_Toc131400291)

[Part 1: Pre-processing file 20](#_Toc131400292)

[Purpose: 20](#_Toc131400293)

[Known trade-offs or possible shortcomings: 20](#_Toc131400294)

[Part 2: DNA Machine Learning file 20](#_Toc131400295)

[Purpose: 20](#_Toc131400296)

[Known trade-offs or possible shortcomings: 20](#_Toc131400297)

[Part 3: Automated HEA 21](#_Toc131400298)

[Purpose: 21](#_Toc131400299)

[Known trade-offs or possible shortcomings: 21](#_Toc131400300)

[Part 4 Calculating baseline carbon emissions 21](#_Toc131400301)

[Purpose: 21](#_Toc131400302)

[Method: 21](#_Toc131400303)

[Assumptions 22](#_Toc131400304)

[Known trade-offs or possible shortcomings: 22](#_Toc131400305)

[Part 5 - producing the automated report 22](#_Toc131400306)

[Section 4: Subsequent modelling of the introduction of a single new clinic site from a list candidate locations 23](#_Toc131400307)

[Acknowledgements: 23](#_Toc131400308)

[Purpose: 23](#_Toc131400309)

[High level pseudocode overview: 23](#_Toc131400310)

[Requirements: 24](#_Toc131400311)

[Method 24](#_Toc131400312)

[Imputing options available to the user 25](#_Toc131400313)

[The user selected lsoa weight adjustment rate – explained 26](#_Toc131400314)

[Outputs 26](#_Toc131400315)

[Explanation of the ‘creating modelled patients’ 27](#_Toc131400316)

[Section 5: Brute force model to map *all* possible candidate locations (for when you don’t have a shortlist of possible locations already) 30](#_Toc131400317)

[Acknowledgements: 30](#_Toc131400318)

[Purpose: 31](#_Toc131400319)

[Requirements / suggested workflow: 31](#_Toc131400320)

[Method 31](#_Toc131400321)

[Parameters used: 31](#_Toc131400322)

# Pre-amble

## What is the SHINE tool and what does it do?

SHINE stands for **S**emi-automated, **H**ealth **I**nequalities, **N**on-admissions, and **E**missions. Each of these areas (health inequalities, non-attendance analysis and carbon emission modelling) was the initial focus of the work. Additional tools/programs have since been developed (explained in this document) continuing one or more of these key areas of focus.

## Why was SHINE built?

SHINE has been produced as a result of the project phase of the Health Service Modelling Association course, run by PenARC, following a conversation between Matt Eves and Anya Gopfert. Both Matt and Anya worked in public health and while both were interested in both the inequalities agenda and sustainability agenda, they each had a respective main area of focus out of the two. During the initial discussion both recognised that, up to now, both agendas are often discussed separately, and agreed there would be value in bringing these two together to support decision making.

## Clarifying terms

Since completing the initial pilot work at the point HSMA concluded, additional programs have been produced to add functionality (to inform decision making when considering introducing a new clinic for a service). All tools continue to be referred to as SHINE, collectively.

## Referencing / Attribution / Branding:

A logo has been produced associated with the SHINE tool, which is included in the git repository (in the “logo” subdirectory). When using / presenting / communicating the outputs from the tool, please incorporate this logo and appropriately reference / attribute the use of the SHINE tool and its authors. For example:

*SHINE tool (2023), produced by: Matthew Eves, Anya Gopfert and Sally Brown*

## License for use:

The SHINE tool has been created with the intention of enabling health systems (where LSOAs units of geography apply) to undertake the analyses in scope at pace and scale and overcome any capacity / capability barrier(s) where these currently exist.

The SHINE tool can be used open source via the MIT license. For further details, please refer to the license terms contained within the LICENSE sub-directory in the repository.

## Acknowledgements

### For development Q&A and trouble-shooting

The authors would like to acknowledge the following organisations and people for their contributions to trouble shooting over the course of the models’ production:

* The PenARC HSMA Team (esp. Dan Chalk for mentorship and clarification, Mike Allen for his excellent machine learning titanic notebooks, and Elliot Coyne & Kerry Pearn for their support with geospatial queries)
* All employing organisations:
  + Derbyshire Community Health Services NHS FT
  + The University of Exeter, and
  + West Sussex County Council
* Previous HSMA alumni consulted and everyone who helped clarifying questions etc.
* The CodeWith online community, in particular Drew Morgan
* Peter Warner-Medley
* ORS for their excellent open-source API

### Data sources:

The tool uses publicly available data sources. The source and nature of the data used, along with a link to the source, statements of acknowledgement (in accordance with the sources’ stated requirements), are listed below. It is your responsibility as a user of this tool to ensure appropriate acknowledgement of the below organisations when sharing SHINE outputs.

* The Office for National Statistics (ONS) (under the Open Government Licence)
  + LSOA level population estimates by single year of age and sex ([link](https://www.ons.gov.uk/peoplepopulationandcommunity/populationandmigration/populationestimates/datasets/lowersuperoutputareamidyearpopulationestimates))
* NOMIS (Official Census and Labour Market Statistics) operating on behalf of ONS (under the Open Government Licence)
  + Ethnicity population estimates for unitary and upper tier authorities ([link](https://www.nomisweb.co.uk/home/copyright.asp))
* Data.gov.uk (under the Open Government Licence)
  + Population weighted centroids ([link](https://www.data.gov.uk/dataset/a40f54f7-b123-4185-952f-da90c56b0564/lower-layer-super-output-areas-december-2011-population-weighted-centroids))
* Ministry of Housing and Communities (under the Open Government Licence)
  + LSOA level index of multiple deprivation data set 2019 ([link](https://www.gov.uk/government/statistics/english-indices-of-deprivation-2019), also available via [this link](https://opendatacommunities.org/resource?uri=http%3A%2F%2Fopendatacommunities.org%2Fdata%2Fsocietal-wellbeing%2Fimd2019%2Findices))
* Open Data Communities (under the Open Government Licence)
  + IMD Decile 2019 ([link](https://opendatacommunities.org/data/societal-wellbeing/imd2019/indices))
* Open Geography Portal (under the Open Government Licence)
  + Boundary files (LSOA, Upper Tier / Unitary LA) ([link](https://www.ons.gov.uk/methodology/geography/licences))
    - Source: Office for National Statistics licensed under the Open Government Licence v.3.0
    - Contains OS data © Crown copyright and database right [2022]
  + Look up files:
    - Files ([link](https://geoportal.statistics.gov.uk/search?collection=Dataset&sort=-created&tags=all(LUP_LSOA_UTLA))):
      * LSOA to CCG and STP look up in England and Wales (April 2020)
      * Lower Tier Local Authority to Upper Tier Local Authority April 2019 England and Wales (April 2019)
      * LSOA to Upper Tier Local Authorities 2021 look up in England and Wales
    - Source: Office for National Statistics licensed under the Open Government Licence v.3.0

### Branding development

The early initial concept for the branding was generated via interaction with Open AI’s DALL-E platform. The initial rough concept was evolved by the project team and subsequently refined by Derbyshire Community Health Services NHS Foundation Trust’s Communications department, who produced the final versions of the branding images.

## How to get in touch with feedback / suggestions:

While every effort has been made to ensure accuracy and transparency of the methods and assumptions adopted for each component of the SHINE tool, it is acknowledged there may be errors and / or areas where further development would be beneficial.

If you have identified an issue or area for improvement, please get in touch (!). Feedback is welcome and the team can be contacted via twitter at:

* @Matt\_Eves\_ ​
* @AnyaGopfert ​
* @Sally\_C\_Brown

# Section 1: Technical Set-Up

## Setting up the environment from the yaml file

The files require a number of python libraries to run. The yaml file titled equity\_carbon\_tool.yaml can be used to set up an “equity\_carbon\_tool” environment containing all required libraries/packages, with compatible versions.

This yaml file is included in the files you have downloaded from github, located in a subdirectory as follows:

* main\_directory
  + yaml\_file [sub\_directory]
    - equity\_carbon\_tool.yaml [file]

For an overview of how to set up an environment please refer to this video by HSMA: <https://youtu.be/fZ2gQZwT1Gw?t=468>

## Folder Structure Required:

The bulleted list below indicates the folder set up required for the code to run. Blue text indicates a folder name, black text indicates a file name.

* Main\_directory\_folder:
  + template.doc
  + all x5 .py code files
  + Raw data
    - govuk
      * LSOA\_Dec2011\_Pop\_Weighted\_Centroids.csv
      * LSOA\_Dec2011\_Pop\_Weighted\_Centroids.geojson
    - local\_service\_data\_test
      * *[save the service data to be used in this sub-directory. Note: it cannot be stored in a folder within this folder and MUST be saved as a .csv file]*
    - ministry\_of\_housing\_communities\_local\_gov
      * Indices\_of\_Multiple\_Deprivation\_(IMD)\_2019.csv
    - nomis\_web
      * nomis\_ethnicity\_unitary\_county.csv
    - ons\_population\_data
      * 2020\_females\_pop\_lsoa\_syoa.csv
      * 2020\_females\_pop\_lsoa\_syoa.xlsx
      * 2020\_males\_pop\_lsoa\_syoa.csv
      * 2020\_males\_pop\_lsoa\_syoa.xlsx
      * 2020\_persons\_pop\_lsoa\_syoa.csv
      * 2020\_persons\_pop\_lsoa\_syoa.xlsx
    - open\_data\_communities
      * imd2019lsoa\_decile.csv
    - open\_geography\_boundary\_files
      * LSOA\_shapefiles
        + Lower\_Layer\_Super\_Output\_Areas\_(December\_2011)\_Boundaries\_Generalised\_Clipped\_(BGC)\_EW\_V3.cpg
        + Lower\_Layer\_Super\_Output\_Areas\_(December\_2011)\_Boundaries\_Generalised\_Clipped\_(BGC)\_EW\_V3.dbf
        + Lower\_Layer\_Super\_Output\_Areas\_(December\_2011)\_Boundaries\_Generalised\_Clipped\_(BGC)\_EW\_V3.prj
        + Lower\_Layer\_Super\_Output\_Areas\_(December\_2011)\_Boundaries\_Generalised\_Clipped\_(BGC)\_EW\_V3.shp
        + Lower\_Layer\_Super\_Output\_Areas\_(December\_2011)\_Boundaries\_Generalised\_Clipped\_(BGC)\_EW\_V3.shx
        + Lower\_Layer\_Super\_Output\_Areas\_(December\_2011)\_Boundaries\_Generalised\_Clipped\_(BGC)\_EW\_V3.xml
    - open\_geography\_portal\_lookups
      * Lower\_Layer\_Super\_Output\_Area\_(2011)\_to\_Upper\_Tier\_Local\_Authorities\_(2021)\_Lookup\_in\_England\_and\_Wales\_.csv
      * Lower\_Tier\_Local\_Authority\_to\_Upper\_Tier\_Local\_Authority\_\_April\_2019\_\_Lookup\_in\_England\_and\_Wales.csv
      * LSOA\_(2011)\_to\_Clinical\_Commissioning\_Groups\_to\_Sustainability\_and\_Transformation\_Partnerships\_(April\_2020)\_Lookup\_in\_England.csv
    - user\_and\_data\_parameters
      * user\_and\_data\_params.xlsx (Note: The user needs to update this file, as relevant for the service (and condition(s) ) they intend to use the SHINE tool for. Once updated, this must be saved in this folder location. What is required in this file is explained later in this document)

## API Key Requirement

In order to use the tool, the user must obtain their own personal API key from Open Route Service at the following address: [Plans | Openrouteservice](https://openrouteservice.org/plans/). ORS is an open source API providing, among many others, the ability to geocode postcodes and calculate the distances between a series of points (both of which are used in the tool).

Please note:

1. Never share your API key or post this online
2. We have made every effort to ensure the required acknowledgements and accreditations have been made to Open Route Service (ORS) in our code and the outputs / report it produces. Please ensure you familiarise yourself with and ensure adherence to all of the ORS acknowledgement requirements *they* require in return for the provision of an API key under their licence terms. This is your responsibility as a user of this tool.

# Section 2: User Guide

## user\_and\_data\_params.xlsx File

### Overview

The program uses the user\_and\_data\_params.xlsx file to enable it to be applied to differing data sets which may have differing column names (**G**ender vs. **g**ender vs. **Sex** etc.) or class labels within the columns (such as **F**emale vs. **f**emale vs. **F** etc.). Enabling scalable application is the main purpose of this file.

### What to complete in the file?

A simple colour-coding key identifies what the user needs to populate as part of the setup process:

* **Yellow** cells: these require an input from the user / need completing for each differing service/use case of the tool. Note: if you intend to use this on *one* service at *different time points* (e.g. repeating an HEA/emissions analysis twice a year), if you intend to keep all parameters the same and there is no change to the source data structure, then after initial setup, there wouldn’t be a need to alter this file for the second run of the analysis).
* **Blue** cells: these require an input from the user at first set up ever, but do not require alteration thereafter for subsequent uses (e.g. using the tool on different services, timeframes of data sets etc). The Open Route Service API Key is an example of this type, which must be completed at initial set up for the tool to run.
* **Green** cells: these *can* be adjusted by the user (by following instructions on the sheet) *if required.* Adjustment is not essential for each use case. Carbon emission equivalent values and proportion travelling by each means are examples of this type. The default values were taken from a .gov site. The default values may be changed if, for example, the user has local insight from a local travel survey that is more recent/relevant (but, crucially, of the same format / type).
* **Grey** cells: These **must not** be changed.

### Structure

The file has the following tabs, each of which requires specific information.

* AssignFieldNamesToVars
* Branding
* DNA\_ML\_parameters
* HEA\_parameters
* HEA\_condition\_details
* Ethnicity\_mapping
* Gender\_mapping
* Carbon\_emissions
* Clinic\_details
* New\_clinic\_details
* emissions\_values
* emissions\_values\_new

What follows is an explanation of each tab in the file to enable this to be completed appropriately and ensure the code will run.

#### AssignFieldNamesToVars

This contains a table where the user must ‘map’ the field names ***exactly*** as they appear in the data set to be used, against the variable names the program uses for those fields. There is an explanation provided for each to help guide you. This ‘mapping’ is required as different services / systems etc. will inevitably record data that semantically and contextually means the *same* thing in *different ways* (e.g. *Gender or Sex* etc.).The mapping here allows the program to be used regardless of how data fields are labelled in the source data.

All fields must be completed except imd\_decile, which is optional. This is because the program gives the option of adding this field to the dataset (based on lsoa\_residence) when it is run.

##### Future development

In a future iteration, with further development of the code, a second table beneath the variable mapping table will require completion. This is anticipated to contain a small number of rows, and will enable the code to run more efficiently by reducing user-interaction and remove hard-coded ‘expected’ values (e.g. Booked for booked appointments), reducing set-up time. This is not yet in place.

#### Branding

This contains 5 drop down selection fields. The first 4 allow the user to choose the colours they wish to use when the code produces charts. An example of where the colour will be used is provided, along with a swatch to show the user the colour they have selected.

The last drop-down field allows the user to select the format of the tables that the code produces in the final report. A preview isn’t available in the excel file itself, but instructions are provided to help the user locate a preview of their selection.

#### DNA\_ML\_parameters

There are two fields in this tab. The first (top\_x) has a default value of 5 (which the user can change). This number will be the number of ‘features’ the Machine Learning models with retrieve that contribute to a higher or lower likelihood of a DNA.

It is suggested that an initial version of the code is run with the default value of 5, and this is varied after that point.

Note: The code may fail to run if:

1. the user enters a number greater than the total number of features present
2. the user enters a number greater than the total number of positive or negative features the model produces

…where the total number of features is the total number of labels of each text field (e.g. *male*, *female*, *transgender*, *non-binary* in the *gender* field, would equate to 4 features).

The second parameter is sample\_size\_for\_ml. This has a default value of 250, and is the size the Machine Learning program will use when running the models. The ML model *may* crash if, by random chance, the samples created only contain one class (DNA *or* Attend), and not *both.* This is because the ML models used require *two classes to be present.* If this happens, it is recommended you increase the sample size parameter entered to a larger number (effectively, increasing the likelihood that when the sample is taken, both classes are randomly sampled).

#### HEA\_parameters

This tab is where the user:

1. selects the Upper Tier or Unitary Local Authority for which they are running the tool from a drop down list,
2. enters a brief text description for the name of the service (this text is used in the final report), and
3. enters a number for the number of equal age range groupings the code will produce (explained below).

The default value for the number of age ranges the code produces is 5. This can present a *small* margin of error in the findings, where the condition(s) being modelled for have an age range that is not equally divisible by 5 (or whatever number the user replaces this with). The model requires whole years for the age ranges it produces, and not decimals (as population data is presented in this way). As such, the logic used identifies the age range between the minimum and maximum age seen by the service, divides this by 5 (or whatever number the user replaces this with) and where this is not a whole number, rounds this down to the nearest whole number of years. It then creates the age ranges by starting at the minimum age and adding the age range interval each time, until it can’t do this without going beyond the max age range value.

**Example**:

A service sees patients from age 16 to 65 and uses the default number of age ranges of 5.

65 – 16 = 49

49 is not equally divisible by 5, as 49 / 5 = 9.8 years.

The code rounds this down to 9 years and then calculates the age ranges to use as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Lower age in calculated age range** | **Calculation for higher age range** | **Higher age in range** | **Age range** |
| 16 (user entered parameter) | 16 + (9-1) = 24 | 24 | 16 – 24 |
| 24 + 1 = 25 | 25 + (9-1) = 33 | 33 | 25 – 33 |
| 33 + 1 = 34 | 34 + (9-1) = 42 | 42 | 34 – 42 |
| 42 + 1 = 43 | 43 + (9-1) = 51 | 51 | 43 – 51 |
| 51 + 1 = 52 | 52 + (9-1) = 60 | 60 | 52 – 60 |

This means that while the service sees patients from 16-65, and the service dataset would include patients 16-65, the “equal intervals” method used currently would create age ranges ending at age 60. This creates a small margin of 5 years of population size data that are excluded from the model.

#### HEA\_condition\_details

This tab is where the user:

1. enters a whole number for the number of “conditions” the model is being run for (max 20)
2. subsequently enters all required parameters for each of the n conditions the model is being run for in the yellow shaded cells in the bottom table (each parameter is further explained below)

##### Parameters for each condition:

The bottom table in the *HEA\_condition\_details* tab is conditionally formatted, depending on the number entered for the number of conditions being modelled, in the top parameter on this tab. So, if the user enters 2 in the “number of conditions” box, then 2 rows of the bottom table will change to yellow, indicating they require inputs from the user (refer to the colour key higher up this document).

The content entered into this table for the parameters for each condition modelled is what underpins the majority of the automated HEA modelling.

One complete line (all yellow boxes) is required to be completed for each condition being modelled for, in the ‘n’ number of conditions the user has indicated in the top box. All yellow cells must be completed. Ignore grey cells.

|  |  |  |
| --- | --- | --- |
| Parameter | Type of input required | Essential or subject to other inputs |
| Condition Name | A brief (1 word) text label, meaningful to the user, to indicate the condition this line relates to. This must relate to at least one attendance reason in the field in the data set associated with the attend\_reason variable (cell B16 of tab AssignFieldNamesToVars in the user\_and\_data\_params.xlsx file) | Essential |
| Prevalence type | Selection from the drop down list of either:  2. Crude rate  3. No rate – census only  (the age-standardised rate option has not been built into the code yet and should not be used) | Essential |
| Numerator | Whole number reflecting the numerator in the prevalence rate. | Conditional – this is ONLY required if the user selects “2. Crude rate” in the ‘prevalence type’ drop down list. In this case, the numerator cell turns yellow indicating it must be completed. If a prevalence type of “3. No rate - census only” is selected, this remains grey, indicating it must be skipped. |
| Denominator | Whole number reflecting the denominator in the prevalence rate. | Conditional – this is ONLY required if the user selects “2. Crude rate” in the ‘prevalence type’ drop down list. In this case, the numerator cell turns yellow indicating it must be completed. If a prevalence type of “3. No rate - census only” is selected, this remains grey, indicating it must be skipped. |
| Minimum age | Whole number reflecting the minimum age that, according to the service access criteria / specification, the service would accept | Essential |
| Maximum age | Whole number reflecting the maximum age that, according to the service access criteria / specification, the service would accept | Essential |
| Patient gender | Single selection from the drop-down list of either:   1. Persons 2. Males only 3. Females only   (Note: these options are inline with the ONS population estimates by single year of age and gender at the time the code was produced). | Essential |
| Proportion of need this represents (e.g. in the case of sexual health, where operationally you have see 9 negative patients for every 1 positive patient, in this case enter 0.1 here) | Enter a proportion >0 and <= 1 | Essential |
| Proportion of the need that would be expected to be seen by this service (e.g. in the case of sexual health services, if 45% is seen by online services, and this tool is being used for an HEA of clinic-based services, enter 0.55 here) | Enter a proportion >0 and <= 1 | Essential |

##### Explanation of the method for adjusting the prevalence rate:

The program undertakes a number of steps to derive an adjusted prevalence rate (know as a “prevalence multiplier”). These steps are as follows:

1. Check whether a crude prevalence is being used
   1. If so, derive the baseline prevalence rate by dividing the numerator value by the denominator value (e.g. 13 / 100 = baseline rate of 0.13)
   2. If not and a census is being used, assign a baseline rate of 1.0 (for either approach this is used as a multiplier to multiply the appropriate summed population of the appropriate age range / gender by, so in this case, we are multiplying by 1.0, meaning the whole population is in scope at this point)
2. Divide the baseline rate by the “proportion of need this represents value”. This is because it may be the case where prevalence rates are known for the presence of a condition, but a service may need to see more people than that rate would suggest. For example, screening services or sexual health STI services, which may test many people of whom only a subset (e.g. 10%) may have a reactive / positive test).
3. Multiply the result from (2) by the “proportion expected to be seen by this service”. This is because it may be that a service is part of a range of services / access points that patients can go to for the given condition. For example, in sexual health services, certain patients may be able to access services online, and so the model only needs to account for the proportion of the need that the service is likely to see (i.e. excluding the online activity, in the above example).

The formula for this is:

Adjusted\_rate = ((numerator / denominator) / “proportion need represents”) \* “proportion to be seen by this service)

#### Ethnicity\_mapping

This tab requires the user to input the different ethnic groups ***exactly*** as they appear in the service data set into the “service\_data\_categories” column (col A). Then, using the drop-down list in the “high\_level\_categories” column (col B), map each category in the service data in Col A to the (potentially less granular) category on the right hand side Col B. This is because the ethnicity population data was available at the less granular level. The correct mapping of the categories will enable the automated ethnicity comparison in the automated HEA.

#### Gender\_mapping

Similar to the ethnicity tab, this tab requires the user to enter all gender labels used in the service data set in the left hand “service\_data\_categories” column, and then, using the drop down list in the “*High\_level\_categories*” column on the right hand side, assign the male service data field to Male, the female service data field to Female, and any other present gender labels to “currently excluded – population data not available”. This is required because at the time of writing (September 2022) the ONS single year of age population estimates at LSOA are only available for “all persons”, “Males” and “Females”. The correct mapping of the gender categories will enable the automated gender comparison in the automated HEA.

#### Clinic\_details

This tab requires you to enter the details of your current service clinics / access points. The table below summarises the parameters expected and the data type of each:

|  |  |  |
| --- | --- | --- |
| Parameter | Data type | Use |
| clinic\_name | Brief text label for the clinic name that is semantically meaningful for the user (ideally, 1-2 words) | The label will be used to label the clinic points on the generated maps. Long labels, especially those on the eastern side of the map for the given LA, can render beneath the key. For this reason, short labels are advised. |
| postcode | Standard UK postcode to be entered without a space, e.g. DE110AE is fine, DE11 0AE is not. | This is used to *geocode* the clinic location to the central point of the postcode using latitude and longitude. These coordinates are subsequently used in the calculation for modelled patient travel carbon emissions. |
| LocalAuthority | Single select from the drop down list to indicate the Local Authority in which this clinic is situated. | Where a service has clinics spanning more than one Local Authority, this ensures the programme is only using the clinics in the LA for which the tool is being run. |

#### New\_clinic\_details

This tab is used for the optional follow-on file to model/estimate the effect of adding a new clinic on the emissions this may generate through patient travel, and a sensitivity analysis of varying levels of up take by previous “unmet need” patients (the purpose, method, trade-offs and assumptions for this subsequent model are outlined later in this document). This tab has several sections, each are explained below.

First table – new clinic details:

This requires details similar to the Clinic\_details tab, with some additional information, summarised in the table below:

|  |  |  |
| --- | --- | --- |
| Parameter | Data type | Use |
| clinic\_name | Brief text label for the clinic name that is semantically meaningful for the user (ideally, 1-2 words) | The label will be used to label the clinic points on the generated maps. Long labels, especially those on the eastern side of the map for the given LA, can render beneath the key. For this reason, short labels are advised. |
| postcode | Standard UK postcode to be entered without a space, e.g. DE110AE is fine, DE11 0AE is not. | This is used to *geocode* the clinic location to the central point of the postcode using latitude and longitude. These coordinates are subsequently used in the calculation for modelled patient travel carbon emissions. |
| LocalAuthority | Single select from the drop down list to indicate the Local Authority in which this clinic is situated. | Where a service has clinics spanning more than one Local Authority, this ensures the programme is only using the clinics in the LA for which the tool is being run. |
| Annual capacity (no. of appointments) | A whole number representing the annual capacity (number of appointments) the service expects to run from this location. It is recommended using this model for annualised figures. | This is used when modelling new patient attendances at the new clinic, up to this user-defined capacity threshold. |
| DNA\_Rate | A decimal value representing the DNA rate, i.e. 0.2 should be entered if the DNA rate is 20%. | This is used when modelling attendance at the new location, to allow the model to account for DNAs (as not *all* appointments will be used) |

##### Method for controlling for missing LSOAs

The model for new locations uses the actual number of patients seen from each LSOA and at each clinic, to weight the chance that a modelled patient is modelled to attend at the new candidate location. However, it is possible the service did not see a single patient from 1 or more LSOAs in the actual service data. By default, this will mean the weight (chance) of a patient being modelled as being from that zero LSOA is 0 (which may not be reasonable). The user has the option of adjusting the method for how the model handles these zero values using the selection box in this section of the tab and can choose one of the three options:

1. Don't replace zeros (retain value of 0)
2. Replace zeros with a single patient (value of 1)
3. Replace zeros with the nearest LSOAs values

These are described in greater detail in section 4.

#### Carbon\_emissions

This has one cell requiring input from the user. The user must enter their own API key into this cell which can be obtained from Open Route Service at the following address: [Plans | Openrouteservice](https://openrouteservice.org/plans/)

Please note:

1. Never share your API key or post this online
2. We have made every effort to ensure the required acknowledgements and accreditations have been made to Open Route Service (ORS) in our code and the outputs / report it produces. Please ensure you familiarise yourself with and ensure adherence to all of the ORS acknowledgement requirements they require in return for the provision of an API key under their licence terms. This is your responsibility as a user of this tool.

#### emissions\_values

It is not essential to change the content of this tab however the user should familiarise themselves with the content.

The content from this tab is taken from the .gov site and contains the average proportion of people that travel via bike, bus, car, motorbike, train, or walking, along with the mean CO2e in units of kg per mile, for each of this means of travel.

If the user has more recent or local data of this sort (e.g. from local travel surveys) the content can be updated to reflect this. However, the structure (column headers and number of columns) must remain unchanged. The user can add new rows immediately beneath the current rows, if additional travel means are required.

#### emissions\_values\_new

refined approach for calculating emissions, which seeks to incorporate a changing proportion of people traveling by each means, depending on the journey distance.

## Explanation of the separate files:

There are 8 files in total, serving 3 broad purposes, summarised in the table below:

|  |  |
| --- | --- |
| Current file names | Purpose and how to run |
| stage\_01\_master\_file\_v003.py  stage\_02\_Objective\_0\_PreProcessing\_003.py  stage\_03\_Objective\_1a\_ML\_v003.py  stage\_04\_Objective\_1b\_HEA\_unmet\_need\_v004.py  stage\_05\_Objective\_2\_carbon\_emissions\_v003.py  stage\_06\_Objective\_2a\_mapping\_file\_v003.py | Running stage\_01\_master\_file is all that is required to run all these files in-turn. The master file “triggers” them to run in order. The user can choose whether to run the ML model (recommended first time, but **pay careful attention to the ROC AUC metric to determine whether the results carry any weight**)  ***Purpose of these files:***   * Stage\_02: clear and process data * Stage\_03: run ML models * Stage\_04: run health equity audit * Stage\_05: quantify emissions estimate * Stage\_06: produces maps |
| Stage\_07\_add\_new\_site\_v06\_final.ipynb | This is a jupyter notebook file which provides various insights when planning a new clinic location (impact on access for the whole LA and the 20% most deprived LSOAs and CO2e emissions generated).  This file requires stages 01-06 to have been run first. |
| Stage\_08\_brute\_force\_optimisation\_iso\_accessibility\_notebook\_03\_final | Stand-alone jupyter notebook file to identify areas to consider for shortlisting new locations.  This can be run before Stage\_07, if you don’t have a specific shortlist of locations in mind.  Caution: this file can take a long time to run (2+ hours) |

## Explanation of what the user must do when running the program:

### Stage\_01 to stage\_06 (HEA, ML DNA, Emissions):

After checking that all the files are set up in the directories as outlined at the start of this document, and that all .py files (the components of the tool) are stored in the same folder at top of the directory tree, the user of the tool needs to run the master .py file. This master file will “call” the other programs in-turn, making them run as required. The user has the option of choosing whether or not to run the machine learning model. It is recommended this is run for the first time at least. On subsequent runs of the model with the same data set, for the same service (e.g. where adjusting one parameter for sensitivity analysis, and then re-running the model) the user should be able to choose not to run the ML models. Depending on the size of the data set, this could save significant run time.

In testing, for a c.40,000 row data set, with 2 conditions, running all files took between 6 and 10 minutes.

The user is required to interact with the program twice. Firstly, to map the attendance\_reason to the condition name the user has entered and secondly, to remove any attendance status other than the two that semantically mean “attended their appointment” and “did not attend their appointment”, respectively.

Once the program has finished running, the user should be able to open the automated report the program has produced. This will be saved in the same location / directory in which the .py files were run from.

In addition, to help a user track back to see what the program has produced during the automated analysis, the program stores all outputs (assets) in the assets\_produced\_by\_code subdirectory.

# Section 3: Explanation of methods and known trade-offs

## Part 1: Pre-processing file

### Purpose:

To read in raw data and enable the user to quickly “clean” this up, and produce 3 processed data sets, ready for use in the subsequent components of the tool.

The program will identify any fields with missing data and give the user the option of removing these fields from the analysis or retaining them and imputing the missing data.

If the user decided to retain the fields with missing data and impute values for these, the method is as follows:

* For text / string data the null / missing value is replaced with the word “missing”
* For numeric (integer or float) values the null / missing values is replaced with the median value of that field.

This imputing stage is required because the machine learning models cannot be run on incomplete data.

### Known trade-offs or possible shortcomings:

If the user decides *not* to impute the values for the field containing missing values, this *whole field* is removed from the processed data set and not passed to the model. Experimentation / further testing is required to determine the effect(s) of this.

At present, the user doesn’t have the option to *choose* which method(s) to apply for imputing missing values (though this has been put in place for subsequently modelling of a new clinic location – see Section 4).

## Part 2: DNA Machine Learning file

### Purpose:

To run a series of variations of a Logistic Regression model on the processed data, to attempt to identify factors predictive of someone not attending their appointment. The number of factors identified is determined by the parameter entered by the user (default is 5).

**Note**: the auto report produced by the program includes a metric called ROC AUC. This indicates how well the model performed. It is important that consideration is given to this metric when interpreting the results of this stage.

### Known trade-offs or possible shortcomings:

Although 9 variants of a model are run, they are all still variations of s logistic regression model. Future development could incorporate alternative predictive models.

In addition, the data set the program has been built around has been intentionally kept small / simple in terms of the number of fields in use. These fields are predominately the protected characteristics plus some basic appointment details. The reason for this was to try to ensure the models could be applied generally to many services. However, this may mean the data set used for the service this model is applied to may not contain sufficient information for any of the 9 variants to perform well / identify predictors robustly. Future development could incorporate additional data fields.

## Part 3: Automated HEA

### Purpose:

This part of the program runs an automated health equity audit. This uses public reference data sets from the ONS, NOMIS Web, and the .gov website.

The user parameters are used here to transform the reference data sets and subset these to just the local authority of interest that the program is being run on, along with population estimates for the gender and age range in accordance with the user’s parameters.

### Known trade-offs or possible shortcomings:

The current method uses a local authority as the minimum level of geography to ‘bound’ the reference data. Future development could be to incorporate alternative reference geographies, such as Primary Care Network, Place or District authority, for example.

The method removes out of area attendances from the data set. This is because the reference population is created to align to the local authority of interest and cannot control for ‘n’ additional local authorities who may be represented in part in the data set due to their having out of area attendances present. This method ensure we are comparing comparable data sets (apples and apples).

## Part 4 Calculating baseline carbon emissions

### Purpose:

This part of the program derives the estimated approximate carbon dioxide equivalent emissions (CO2e) for patients travelling to and from face-to-face appointments.

### Method:

This section of the program makes use of the [Open Route Service](https://openrouteservice.org/) geocode and routine APIs, using the user’s API key (this needs to be sought separately from the ORS website).

The code first reads in the information from user\_params file relating to the current clinics. The postcodes for each clinic are ‘geocoded’ via the ORS geocode API. This retrieves the latitude and longitude for the central point of the postcode. This is used as the approximate location for the clinic.

The population weighted centroids are read into the file from a .gov .csv file (LSOA\_Dec2011\_Pop\_Weighted\_Centroids.csv).

Now the program has a set of coordinates for the starting point (population weighted centroid for each LSOA) and a set of coordinates for each current clinic location.

This allows a distance matrix to be produced consisting of the distance from every LSOA population weighted centroid to every geocoded clinic location.

An identically structured patient matrix is created, only instead of containing the **distance** from each LSOA, it contains the **number of patients** that attended from each LSOA that attended at each respective clinic location.

Because these two matrixes are identically structured, the program can multiply the two matrixes together. This produces the total approximate distance travelled to appointments. This distance is then doubled, as we assume patients make a return journey home after their appointment.

The program then sums up the number of miles in the resulting matrix, to give a single figure (the number of miles travelled in total to and from each appointment, across all clinics, for attended appointments only).

### Assumptions

The model doesn’t model journey distance exactly. It doesn’t use each individual patients’ home address. Instead, the model uses the population weighted centroid of the LSOA that the patients resident address lies within. The population weighted centroid is the most populous location in the LSOA and is available via reference tables published on the UK .gov website. Essentially, this models all residents as living in the same location (e.g. a block of flats) in the same location on the map. This assumption introduces a margin of error, in that patients will not *all* live on or immediately next to the population weighted central location of the LSOA. This error is bounded though because the patients ***will***live within the LSOA ***somewhere***. The trade-off was that this removes the clear IG consideration of modelling individual addresses. In addition, modelling tens of thousands of individual locations to the clinics would exceed the Open Route Service’s free API usage cap, which would limit the usability of the tool created. Furthermore, ***even if there was no limit on the ORS API usage***, modelling so many routes would (and in all likelihood, significantly) increase the run time of the model / processing power required to run the model.

The model does not use the exact destination location for each clinic (it uses the approx location based on the centroid of the postcode location). This has a similar impact to the assumption above of using the population weighted centroid of the LSOA location, only the margin of error is likely smaller, ass the postcode area is smaller. The trade-off benefits are the same.

### Known trade-offs or possible shortcomings:

A key trade-off is that the model assumes all residents of a given LSOA all reside in the same location.

## Part 5 - producing the automated report

Once all components of the model / program have been run, the model then runs code to produce an automated report, consisting of all the outputs as applicable to the service / dataset that has been used.

This incorporates all findings from each stage of the code already run by this point.

The report is structured / worded in a way that clearly shows the user the findings from each stage of the model, and the assumptions that have been made at each stage. The intention is that this can be used with relevant stakeholders for the service, to inform next steps (action planning and decision making).

**Note**: All outputs that are produced by the code (charts, maps, data tables, figures etc. are saved to the local computer’s directory in a sub-directory to the current working directory where the code is being run from. These are stored in a sub-directory folder called “assets\_produced\_by\_code”.

# Section 4: Subsequent modelling of the introduction of a single new clinic site from a list candidate locations

## Acknowledgements:

This section was undertaken as a subsequent development to the HSMA course. It was made possible due to the healthier futures action fund funding provided by The Greener NHS. This standalone model can be run after the original HSMA program has been run. This is because the original HSMA program (covered in sections above) pre-processes the data and produces assets required as inputs to this additional (follow-on) model.

## Purpose:

The purpose of this program is to support decision making when considering where to locate a new clinic location, if several candidate locations are available / under consideration.

The outputs produced include:

1. Sensitivity analysis for the number of patients attending who hadn’t previously attended (unmet need)
2. Modelling an estimate for the *additional CO2e emissions the introduction of each candidate location could introduce*
3. Quantify the net change in the % of the geographic area for the whole local authority and the subset area made up of the most deprived 20% LSOAs only, for 4 travel means (driving by car, cycling, walking, wheelchair travel), for each candidate location being considered.
4. An automated output report summarising all findings in a standardised fashion.

## High level pseudocode overview:

1. Read in distance matrix produced in pre-processing
2. Read in patient matrix produced in pre-processing
3. Create a new matrix containing distances from the LSOAs to each new candidate clinic location
4. Read in parameters from the parameters file, create global params Class
5. Review the patient matrix, check whether at least 1 patient attended from each LSOA (regardless of where they attended)
6. If there are LSOAs with 0 patients, interpolate these rows using the user-selected method (value of 0, value of 1 or the values from the nearest non-zero LSOA, determined using haversine distance in miles between two points)
7. Check whether each new location is closer to the LSOA centroid than the previous shortest journey to the closest ACTUAL clinic
8. If it is, increase the weighted patients attending from that LSOA by X% for every km the potential NEW journey to the candidate location is reduced by, relative to known actual shortest journey
9. Create Lsoa Class and New Random Patient Class (for modelling made-up patients weighted in accordance with the known patient profile re geography / clinic accessed)
10. Run the model: produce modelled (made-up) patients based on the known weights derived using actual data with adjustments above.
11. Quantify net change in the accessibility % of the LA of interest that can access at least 1 clinic by car, bike, cycle or wheelchair within the user-provided threshold journey times (in minutes)
12. Quantify net change in the accessibility % of the core 20% subset of LSOAs in the LA of interest that can access at least 1 clinic by car, bike, cycle or wheelchair within the user-provided threshold journey times (in minutes)
13. produce automated report summarising findings from the model

## Requirements:

The user must run the model described above first, via the master .py file. This is because the initial code produces assets / outputs which are used as *inputs* for the subsequently developed code to model the introduction of a new clinic site.

## Method

The code initially reads-in as inputs the following files, produced as outputs assets from the preceding master .py file:

1. Distance matrix
2. Patient matrix

In addition, the following user parameters are read from the New\_clinic\_details tab of the user\_params file into variables:

* Local authority of interest (in which the current clinics are based, and the data set relates to)
* apiKey – the users’ API key for the Open Route Service API
* the new candidate clinic locations. Where these have more than one local authority of residence present, these are subset to just the Local Authority of residence of interest. These include the number of appointments being modelled per year for each new candidate location and the DNA rate to use in the model. To standardise outputs it is recommended these are kept the same for all sites being modelled (though it is accepted this may not be your use case, in which case it is fine to adjust according to your needs)
* The users’ chosen method for how to handle zero LSOA values (please see the table below for an explanation of each option)
* The user entered threshold journey time in minutes for the 4 means of travel included in the model (walking, cycling, driving by car, or wheelchair travel). These must be whole numbers. **Note**: in testing an anomaly was found when entering a travel time of 30 minutes for one means of travel. If this occurs users are advised to adjust their travel time and re-try running the model.
* The existing clinic locations. As above, where these have more than one local authority of residence present, these are subset to just the Local Authority of residence of interest.
* The modelled met / unmet need figures (though despite being read-in, these are currently not used – it is for the user whether this would be appropriate for their use case. If this is to be used as one of the scenarios, the user will need to “uncomment” the code where indicated)
* The number of conditions being modelled for
* CO2 emissions information (proportion travelling by each travel means by varying distances)
* The kg emissions per mile for various means of travel
* The number of times to run the model (model runs) - the default value is 10. An approximation is given as to the increase in run time (in minutes) the selected number of runs may cause. This is indicative only and intended to inform the user of the run-time trade-off for increased runs (especially for a large value). The model will run for the number of times the user has entered, calculate the results for each individual run, and then average the results over all runs. The rationale for this approach is because the model is using weighted random allocation of features when modelling patients’ attendance the clinic. It would be wrong to consider the output of the model to be an exact figure, instead it is an indicative figure based on the assumptions and trade-offs mentioned in this guide. Averaging the results should average out the error involved in the calculation.
* The user selected lsoa weight adjustment rate – described below.

### Imputing options available to the user

The user can select one of three options to instruct the program how to handle any LSOAs from which the service has not seen any patients / there are no recorded attendances in the data set being used. These are described in the table below along with an explanation as to the method applied for each.

|  |  |
| --- | --- |
| Imputing method selection | Description of method applied |
| Don't replace zeros (retain value of 0) | The 0 value is not replaced. The model runs with the original (processed) data passed in at the outset. This means the weight (probability) applied to these LSOAs when modelling patient attendances to the new clinic remains as 0, which may not be realistic (by way of extreme example, if the clinic is introduced immediately next door to the LSOA centroid, this may increase the likelihood of attendance if the previous nearest clinic was many miles away). |
| Replace zeros with a single patient (value of 1) | All zero values are replaced with a nominal low value of 1 patient. This ensures the probability is not 0 when modelling patients and assigning them to a resident home LSOA. It also ensures the model can subsequently model increased likelihood of attendance where the new location is closer to the home address (proxy of LSOA centroid) than the distance to the actual current clinic (described in more detail below) |
| Replace zeros with the nearest LSOAs values | Like the above method for replacing all 0 values with 1, however this time, the model first locates the nearest neighbouring non-zero LSOA to each zero LSOA. This is done by calculating the haversine distance between each zero LSOA centroid and every non-zero LSOA centroid, then finding the minimum distance, and retrieving the associated LSOA code associated with that minimum distance. Next, the program retrieves the actual attendance values associated with that nonzero closest neighbouring LSOA and using these values, replaces the zero values in the zero LSOA. |

### The user selected lsoa weight adjustment rate – explained

The user can enter the rate the model will use to increase modelled attendances by, relative to how much the modelled distance to the clinic has reduced by, compared to the distance to the closest actual current clinic. For example, the user may enter 1.5% per mile. In this situation, if the new clinic represents a reduced journey distance compared to their current shortest distance to their nearest actual clinic, for every mile their journey is reduced by, the weight (chance) the model will use when creating modelled patients to attend the new clinic will be increased by 1.5%. For example, if the closest current clinic is 10 miles away and the new candidate location is 5 miles away, the net reduction in distance is 5 miles. The code will convert this to km by multiplying by 1.6 and rounding the result. It will then complete a for-loop, looping over each km in the total rounded km’s, and for each km (iteration of the loop) increase the weight by the user-provided %, illustrated below:

Net\_reduction\_in\_miles = 5

Net\_reduction\_in\_km = 5 \* 1.6 = 8

For km in net\_reduction\_in\_km:

Weight = weight \* 1.015

Note: the user has the option of entering their parameter in units of miles or Kilometres.

### Outputs

The model includes the following key outputs:

1. A sensitivity analysis modelling the:
   1. Approximate number of patients not having previously attended, attending the new location
   2. Approximate estimate of CO2e emissions the addition of each candidate clinic may introduce as a result of patient travel to / from appointments

… presented as an x-y scatter chart, for each scenario in the sensitivity analysis.

1. The % of the Local Authority that can access the baseline as-is clinics by each of the 4 means of travel, within the user-provided journey time thresholds, and the net % increase as a result of introducing each candidate location
2. As above for (2), but for just the core 20% most deprived LSOAs in the Local Authority area
3. Isochrone maps plotting the current clinics, the new candidate clinic location, and the isochrone of the area that could access at least 1 clinic within the user-provided X minutes by each of Y travel means.

### Explanation of the ‘creating modelled patients’

The model using randomly allocated features to patients, using known data to inform the weight/chance of each random allocation. The method used for the model is explained below.

New patients are modelled using the New\_Random\_Patient\_experimental Class. This Class produces patient Objects, which have the following attributes:

|  |  |
| --- | --- |
| Patient object attribute | Description |
| modelled\_lsoa | This is the modelled patient’s lsoa of residence, randomly selected using weights as follows:   1. The number of patients actually attending from each LSOA 2. Adjusted for zero LSOAs in line with the user-entered method, and 3. Adjusted where the new location presents a shorter journey distance according to the user-entered parameter |
| dna\_rate | The modelled dna rate, used as a threshold value to determine whether or not the patient is modelled as a DNA or an attended appointment |
| clinic\_attended\_before | The clinic the patient is modelled to have attended before. This is initially assigned to all patients in the model, but is only present in the final dataframe for patients modelled to have previously attended.  The method of assigning the legacy clinic uses the random.choices method, using the number of patients who attended each actual current clinic from the LSOA the modelled patient has been assigned to as the weights for the random selection. |
| attend\_new\_location | Indicates whether this patient should attend the new location. This is determined based on whether the distance to the new candidate location is *less than* the distance to the modelled “clinic\_attended\_before”. If the distance *is* less, then a new modelled patient (object) will be created. If it isn’t the loop will continue selecting with random weighted chances until this is condition is met. As such, this value will be True for *all* patients included in the model output as the decision re whether to attend is handled before the patient object is instantiated. |
| attended\_before | Boolean value to indicate whether the patient has attended before (True = has attended before). This is a random selection using the sensitivity weights (20%, 40%, 60%, or 80% chance of a new patient being modelled). A whole new iteration of the model for *all* clinics is undertaken for each sensitivity weight. |
| dist\_original\_location | The distance from the LSOA centroid to the clinic the patient is modelled to have attended before |
| dist\_new\_location | The distance from the LSOA centroid to the clinic the current candidate location |
| attended | A Bool value, derived during the creation of the patient object via the new\_random\_patient\_experimental class, by checking whether a random float between 0 and 1 is less than the dna\_rate value (if it *is* less than or equal to the dna rate, the value is set to False, meaning they are modelled to DNA) |

The run model function contains a nested for-for-while loop, as follows:

For new\_patient\_weight\_scenario in list\_of\_new\_patient\_weight\_sensitivity\_options:

For num in range(len(new\_clinic\_location\_names)):

Counter = 0

While counter < capacity for *this* clinic:

<call create patient function>

The process for creating the patient includes a series of random weight selections that takes place *before* the create patient function is called, as follows:

#### Step 1: assign random LSOA

#### Weighted according to:

1. the number of patients the service has seen,
2. which were inflated wherever the new location has a reduced distance to that LSOA than the original shortest distance to the nearest current clinic

*Step 2: assign legacy clinic*

By:

1. retrieving the list containing the number of patients attending each clinic, in order, from the modelled LSOA.
2. Using this as the weights to assign this modelled patient to a historical clinic location.

#### Step 3: check whether distance to the legacy clinic < new clinic.

Assign a Bool value to indicate whether the made-up patient is modelled to attend the new location (True) or not (False). The logic is simply if the distance to the new location is less than the distance to the original location they were modelled to have attended previously, then assume they will accept the appointment at the new location (they *may* still be modelled to DNA this appointment at the new location)

#### Step 4: if True, create new random (modelled) patient object

If above is true (distance to new clinic < distance to closest existing clinic), then:

1. Randomly determine whether the patient will be modelled to be a “new patient” (unit of unmet need) or having “attended before”). This weighted allocation is inline with the sensitivity analysis and will be one of 20%, 40%, 60% or 80% chance of being modelled to be a new patient.
2. Create a New\_Random\_Patient\_experimental object by passing modelled LSOA, modelled legacy clinic, modelled new clinic.

#### Step 5: populate a list of appointments with modelled Patient objects

For each patient object created, append them to a list representing the complete capacity for the current clinic the model is working on in the loop at present.

#### Step 6: Populate a nested list of clinic with the list of patient objects produced in Step 5.

When the model concludes, the outer list will have a length equivalent to the number of candidate locations being modelled for, and the inner lists will have lengths equivalent to the capacity (number of appointments) parameters the user entered for each clinic.

#### Step 7: produce a dictionary and a data frame of the modelled patients.

The data frame is created to make subsequent analysis of results easier. Also, it is easier to view the modelled patient profile from the data frame rather than inspecting many different patient objects one by one.

#### Note on speed for this section:

For context, when testing with 3 candidate clinic locations, 200 appointment capacity per clinic location, 4 sensitivity analysis scenarios, over 10 model runs per candidate location, this causes the model to produce a total of 200 x 3 x 10 x 4 = 24,000 modelled patient objects, in < 10 seconds.

Assumptions

The same assumptions were made as in the original baseline emissions code. That is, assuming all modelled patients are resident in the same geographic location, that patients return home after their attendance, that all patients make their own way (no two patients car share), the proportions attending by various travel means are in line with the .gov published proportions, and that the emissions value per mile travelled per means of travel is in line with that of the published emissions values.

Trade off:

The same trade-offs are present here as in the original baseline emissions code

In addition, there is a potential trade-off of imputing a value of 1 or the nearest neighbouring LSOAs values for each zero LSOA, rather than using zero. This is because it may legitimately be the case the number of presentations from the LSOA will remain as zero. While the decision rests with the user of the tool, it was felt appropriate to enable users to impute values because only with a value for all LSOAs of >1 will the model dynamically inflate the weight based on the reduction in distance to the new clinic relative to the previous shortest distance. If zero LSOAs remain with a value of 0 then clearly they will remain as 0 when multiplied by a weighting factor. This means if the user selects to retain zeros as 0, they are effectively also opting out of the dynamic increasing in weighting component of the model too.

Another trade-off in the model is the rate limit of the Open Route Service API. At the time of writing this is set to limited by ORS to 20 api calls per minute. Because of this rate limit, the program initially crashed because the run speed of the program was such that it meant the API was being called at a rate that was too fast than the limit allowed. To solve this problem the program applies some checks to avoid crashing due to a rate limit error. Firstly, by checking for the last run of the isochrone API and the system time when trying to run it again. If the interval is < 1 minute, the program effectively waits / idles until 1 minute has elapsed (effectively resetting the rate limit). Thereafter it introduces a 4 second delay between each API call ending and the next one starting. In testing this has worked well and allowed the program to run start to end without crashing.

Auto-report

Once the file has run, all outputs are summarised into a standardised, automated report.

Example run time, for a local authority with 491 LSOAs, 10 current clinics, 3 candidate locations, and 4 means of travel, is 6.5 minutes. In this example 50% of the run-time is due to enforced delays to avoid the API rate limits being reached (which would crash the program). As such, a further development could be introduced that could explore saving outputs from the code, to read these into the file, avoiding the use of the API for subsequent runs of the model (though the ORS licensing terms / requirements would need to be checked to ensure this is not breaking any terms). This scenario would be particularly useful when re-running the model several times, and adjusting one parameter. So long as this parameter is not the journey time for driving, cycling, wheelchair, walking, this would save 50% of the run time, for each subsequent run of the program.

# Section 5: Brute force model to map *all* possible candidate locations (for when you don’t have a shortlist of possible locations already)

## Acknowledgements:

This section was undertaken as a subsequent development to the HSMA course. It was made possible due to the healthier futures action fund funding provided by The Greener NHS. This standalone model can be run with having already run the HSMA program.

## Purpose:

The purpose of this program is to support the user when they don’t already have a shortlist of candidate locations in mind, by producing heat maps of the net gain in access for the whole Local Authority, as a result of putting the new clinic in each possible LSOA (population weighted centroid), for all LSOAs that don’t currently contain 1 or more clinics.

The outputs produced include:

1. Heat maps for each means of travel the user has run the file for (driving, cycling, walking, or wheelchair travel), showing the net percentage point gain in accessibility for each travel means.
2. An automated output report summarising all findings in a standardised fashion.

## Requirements / suggested workflow:

There is no need for the user to have run any of the previous files. This file is intended to support the decision making for candidate locations (where a short list isn’t already known). It can be run initially, and then, once a short list of sites has been produced, the user can run the master file (stages 01 to 06), and then stage 07 to see the relative trade-offs of the candidate locations (as outlined in the above section).

## Method

* This file adopts a “bruteforce” optimisation approach, and literally works through *every single possible LSOA not currently containing a clinic.*
* For this reason, the file can take a long time to run – in testing, it took approx. 33 mins to run each means of travel, so, if call 4 are included, the file could take >2 hours to run.

## Parameters used:

* The user entered threshold journey time in minutes for the 4 means of travel included in the model (walking, cycling, driving by car, or wheelchair travel). These must be whole numbers.
* The existing clinic locations. As above, where these have more than one local authority of residence present, these are subset to just the Local Authority of residence of interest.
* CO2 emissions information (proportion travelling by each travel means by varying distances)
* The kg emissions per mile for various means of travel