EDASIDA GUI User Guide - Overall

September 2025

Contents

1 Introduction 3

2 Installation 3

2.1 Download and extract the files 3

2.2 Launch the application 3

2.2.1 Handling Windows security notifications 4

3 Getting Started 5

3.1 A working example 5

4 Key Features and Functions 9

4.1 Data Tab 9

4.1.1 Information about the data 10

4.1.2 Variable information 10

4.2 Analytical Outputs tab 11

4.2.1 Output weights 11

4.2.2 Analytical outputs 12

4.3 Main Buttons 12

4.3.1 Validation 12

4.3.2 Running the app 13

4.3.3 Saving output 13

5 Advanced Features 13

5.1 Advanced Settings tab 13

5.1.1 Interactive checkboxes 14

5.1.2 Available settings 14

5.1.3 Restoring the default settings 15

6 Template Guide 15

6.1 Introduction 15

6.2 Data templates 15

6.2.1 Template 01 – Variable names and types 16

6.2.2 Template 02 – Categorical and integer variables 16

6.2.3 Template 03 – continuous variables 16

6.3 Analytical output templates 17

6.3.1 Template 02 – Univariate categorical or integer table 18

6.3.2 Template 03 – Univariate continuous 18
6.3.3 Template 04 – Table: bivariate categorical or integer 18
6.3.4 Template 05 – Table of conditional means 18
6.3.5 Template 06 – Table of conditional medians 19
6.3.6 Template 07 – Table of bivariate conditional means 19
6.3.7 Template 08 – Table of bivariate conditional medians 20
6.3.8 Template 09 – Binary logistic regression 21
6.3.9 Template 10 – Binary logistic regression with categorical predictors 21
6.3.10 Template 11 – Linear regression 22
6.3.11 Template 12 – Linear regression with categorical predictors 23
6.3.12 Template 13 – Correlation matrix 23
6.3.13 Template 14 – Principal component analysis 24
6.3.14 Template 15 – K-means cluster analysis 25
7 Troubleshooting 25
7.1 Introduction 25
7.2 Parameters 25
7.3 Frequently Asked Questions 27
Appendix A – Analytical Outputs Information 29
Introduction 29
Analytical outputs 29
Table of univariate categorical or integer 29
Univariate mean 29
Table of bivariate categorical or integer 29
Table of conditional means 30
Table of conditional medians 30
Table of bivariate conditional means 30
Table of bivariate conditional medians 31
Binary Logistic regression (with/without categorical predictors) 31
Linear regression (with/without categorical predictors) 31
Correlation matrix 32
Principal component analysis 32
K-means cluster analysis 33

1 Introduction

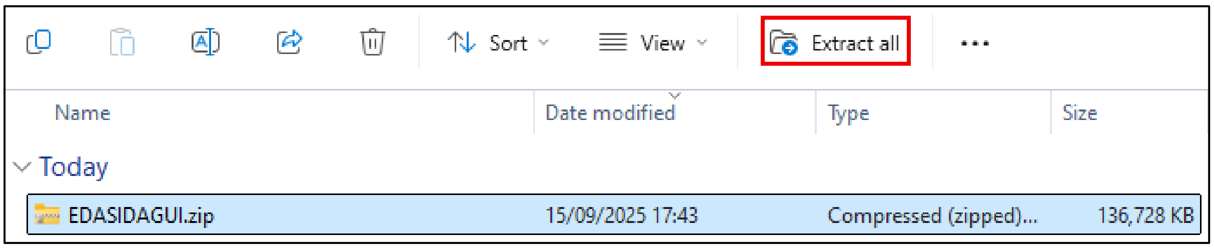
The EDASIDA GUI (Graphical User Interface) is a tool for users to generate synthetic datasets using an Evolutionary Algorithm (EA). This was developed in the Enhancing Data Accessibility and Security through Data Synthesis (EDASIDA) project.

This document provides information on how to install and run the EDASIDA GUI and also covers what to do if there are any problems. The GUI is designed to synthesize data where there is no direct access to the original dataset, instead using information and analytical outputs (such as a linear regression or bivariate table) gathered from the original dataset. The resulting synthetic data should then closely reproduce this analysis.

This method may be particularly useful where datasets are held in restricted settings, such as Trusted Research Environments (TREs). The resulting synthetic data can then be safely shared. Running the EA can be done directly from the command line, but for ease of use the GUI was developed.

2 Installation

2.1 Download and extract the files



Name	Date modified	Type	Size
Today			
EDASIDAGUI.zip	15/09/2025 17:43	Compressed (zipped)...	136,728 KB

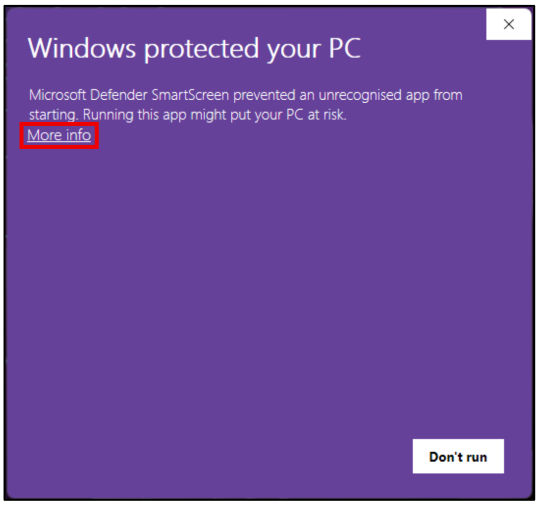
To begin, download the provided "EDASIDAGUI.zip" file and extract its contents to a convenient location on your device. This may take a couple of minutes. After extracting the zip file, open the extracted folder and you should find the application folder ("EDASIDAGUI"), a folder ("example") with examples of the .csv files used in the Getting Started section (section 3 of this guide) section and a folder ("templates") with .csv templates for each of the different input types (see section 6), together with a "README" and "LICENSE" file.

2.2 Launch the application

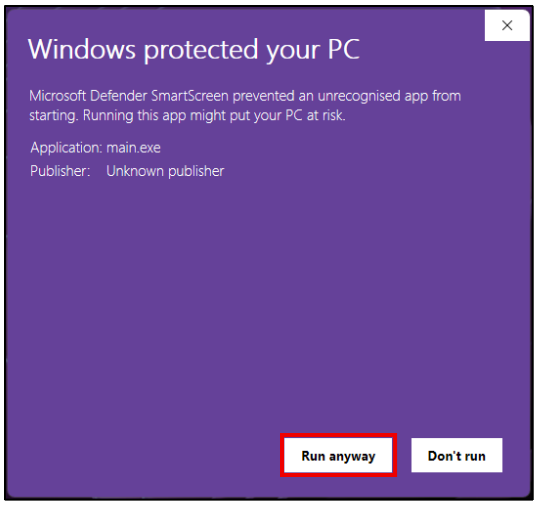
The following instructions are also available in the "README.md" file located in the extracted folder, this can be opened with any text editor such as Notepad.

1. Navigate to the "EDASIDAGUI" folder which is inside the extracted folder
2. Locate the file named "main.exe"
3. Double-click "main.exe" to run the GUI application
4. If you are prompted with a security warning, please unblock the executable before retrying

2.2.1 Handling Windows security notifications


Windows protected your PC

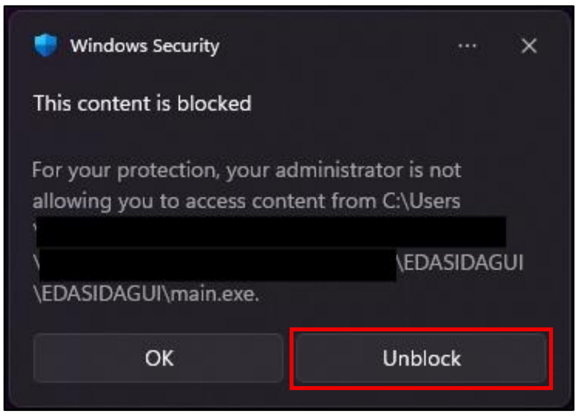
Microsoft Defender SmartScreen prevented an unrecognised app from starting. Running this app might put your PC at risk.


Windows protected your PC

Microsoft Defender SmartScreen prevented an unrecognised app from starting. Running this app might put your PC at risk.

Application: main.exe
Publisher: Unknown publisher

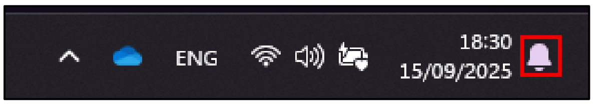
Windows Defender may pop up with a prompt window suggesting it is unsafe to run this application, this can be bypassed by clicking "More info" and then the "Run anyway" button.

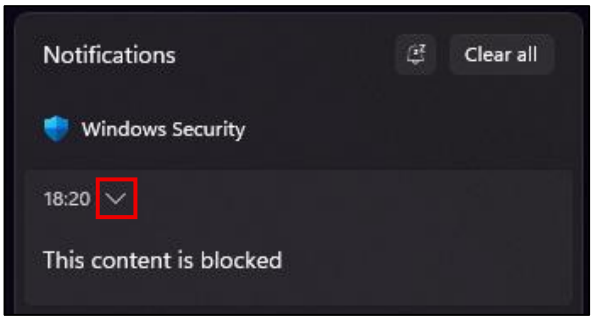

Windows Security

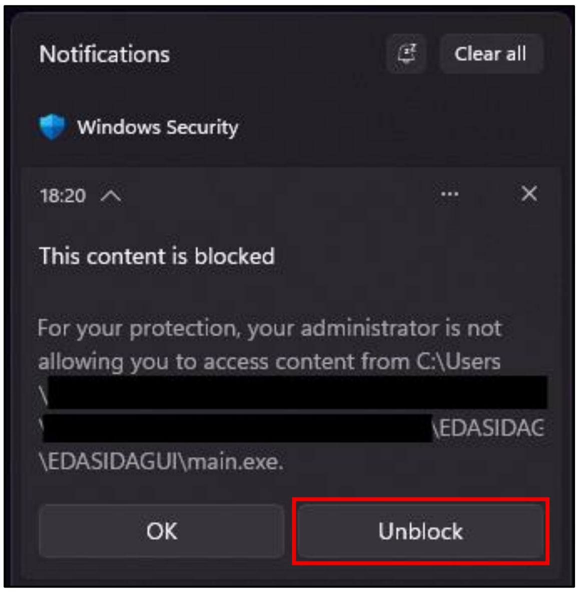
This content is blocked

For your protection, your administrator is not allowing you to access content from C:\Users \[redacted] \[redacted]\EDASIDAGUI \EDASIDAGUI\main.exe.

If you encounter a Windows Security notification like the one shown above, it means that your device's security settings are preventing access to the GUI. If you are happy to do so, please click "Unblock" or "Dismiss" when prompted.







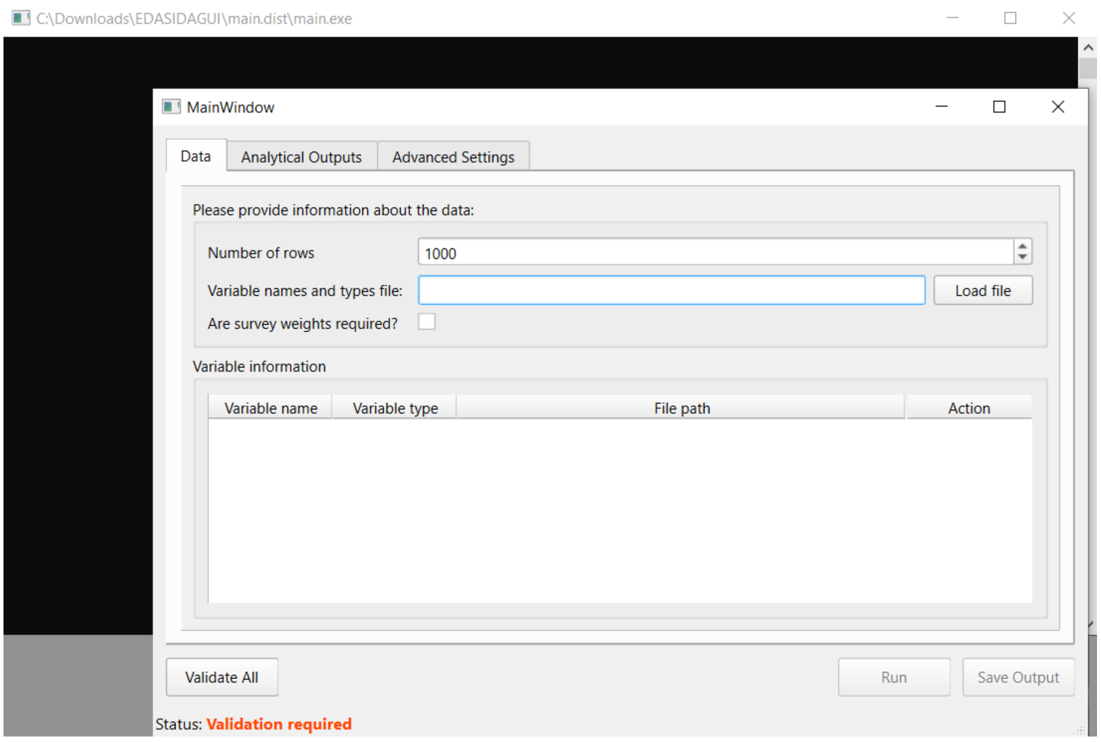
This content is blocked

For your protection, your administrator is not allowing you to access content from C:\Users \[redacted] \[redacted]\EDASIDAG \EDASIDAGUI\main.exe.

If you could not follow the prompt in the short timeframe of this pop up, you may also find it in the notification centre. Double click on the “main.exe” executable again (if required), to launch the application.

Note: All screenshots in this section were captured on Windows 11.

Once launched, the GUI should appear alongside a black box window (this console will be used to display dialogue from the GUI):



Please provide information about the data:

Number of rows: 1000

Variable names and types file:

Are survey weights required?

Variable information

Variable name	Variable type	File path	Action

Status: **Validation required**

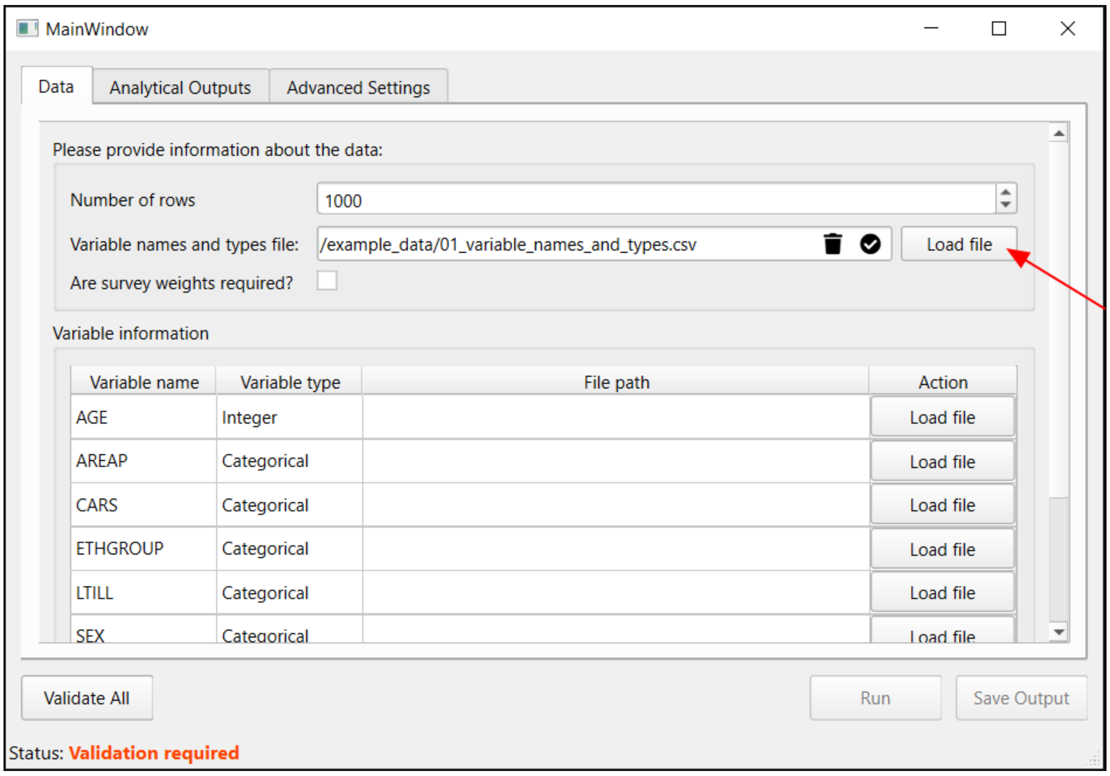
3 Getting Started

3.1 A working example

The EDASIDA GUI requires specially formatted .csv files to provide it with information about the data that you would like to synthesize. Templates of these files are contained in the “templates” folder, and the Template Guide in Section 6 provides information on how to fill them in.

To enable a working demonstration of the GUI, included with the installation files are some example data files. The following steps instruct on how to load these files and perform a run of the EA. This example will load distributional information about 9 variables and an analytical output (of a binary logistic regression) that will drive the production of the synthetic data.

1. On the main Data tab click the “Load file” button, navigate to the “example” folder (you will first need to navigate outside of the current folder back to the original EDASIDAGUI folder then into the “example” folder) and click on the “01_variable_names_and_types.csv” file.
   Loading this file populates the “Variable information” table with each of the variable names (and types) to be synthesized:



Please provide information about the data:

Number of rows: 1000

Variable names and types file: /example_data/01_variable_names_and_types.csv

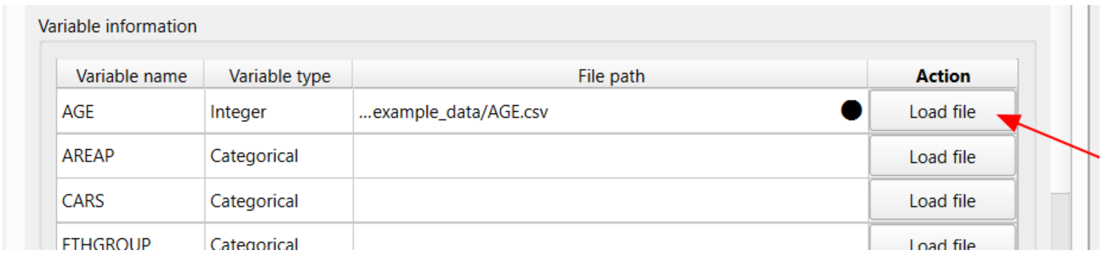
Are survey weights required? ☐

Variable information

Variable name	Variable type	File path	Action
AGE	Integer		Load file
AREAP	Categorical		Load file
CARS	Categorical		Load file
ETHGROUP	Categorical		Load file
LTILL	Categorical		Load file
SEX	Categorical		Load file

Status: **Validation required**

2. For each variable click on the corresponding "Load file" button and navigate to the "example" folder and load the corresponding file. For example, for AGE you would load "AGE.csv".

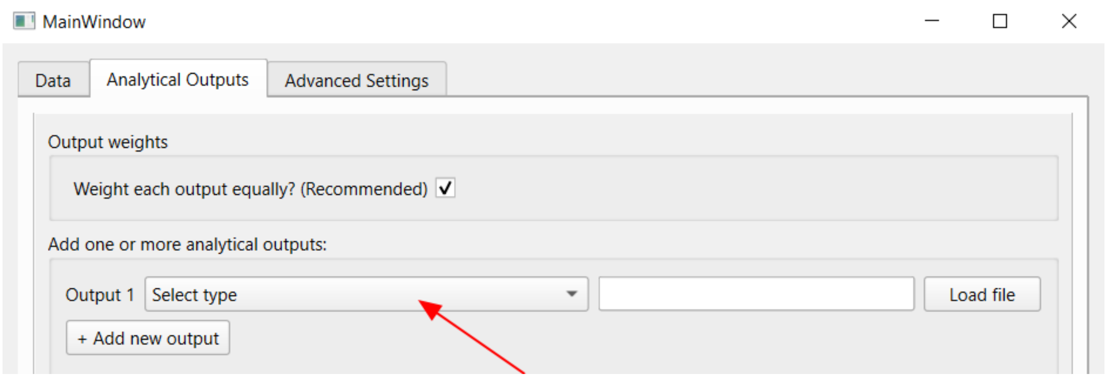


Variable information

Variable name	Variable type	File path	Action
AGE	Integer	...example_data/AGE.csv ●	Load file
AREAP	Categorical		Load file
CARS	Categorical		Load file
ETHGROUP	Categorical		Load file

Do this for all variables – you can scroll down the list if it is not all visible. Once completed all the variable information has been loaded.

3. Navigate to the "Analytical Outputs" tab (by clicking on it) and then click on "Select type" for Output 1.



Output weights

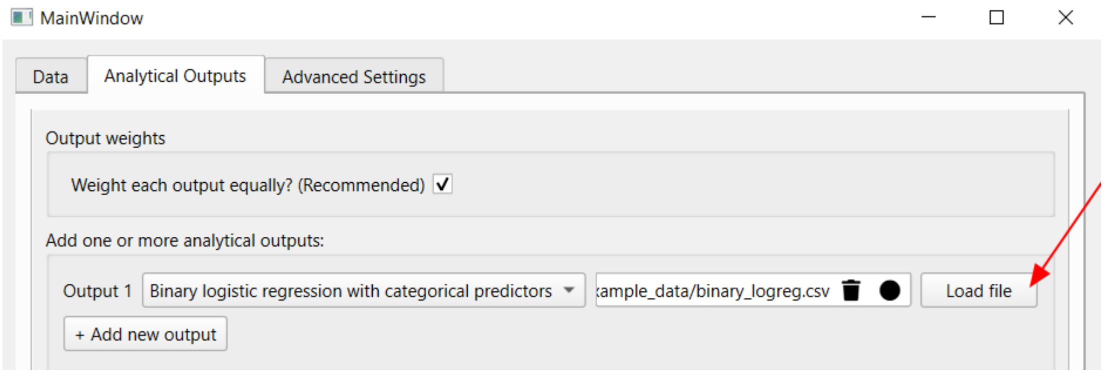
Weight each output equally? (Recommended) ✓

Add one or more analytical outputs:

Output 1: Select type

Choose "Binary logistic regression with categorical predictors" from the drop-down list.

4. Click on "Load file", navigate to the "example" folder and choose the "binary_logreg.csv" file:



MainWindow

Data | Analytical Outputs | Advanced Settings

Output weights

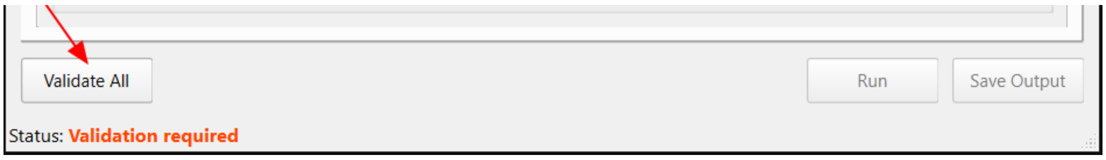
Weight each output equally? (Recommended) ✓

Add one or more analytical outputs:

Output 1 | Binary logistic regression with categorical predictors | xample_data/binary_logreg.csv | Load file

+ Add new output

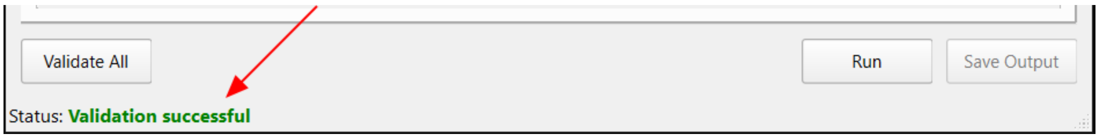
5. Now that all files have been loaded, click on "Validate all", this performs some basic checks on the csv files to ensure they are in the correct format:



Validate All | Run | Save Output

Status: Validation required

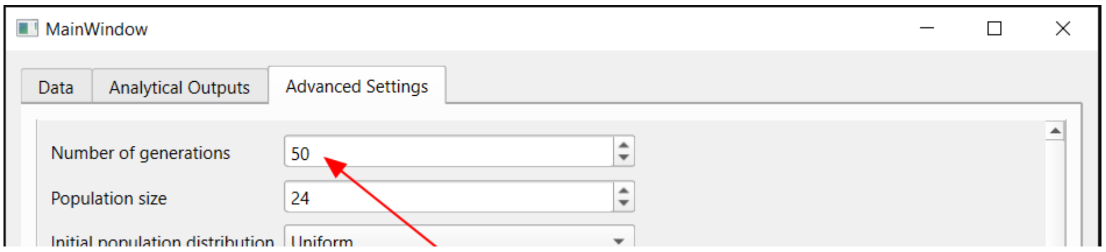
the Status message should change to "Validation successful" if all files were validated:



Validate All | Run | Save Output

Status: Validation successful

6. Click on the "Advanced settings" tab and then set the number of generations to 50 (by default it is 500 but we don't need that many to simply demonstrate the EA):



MainWindow

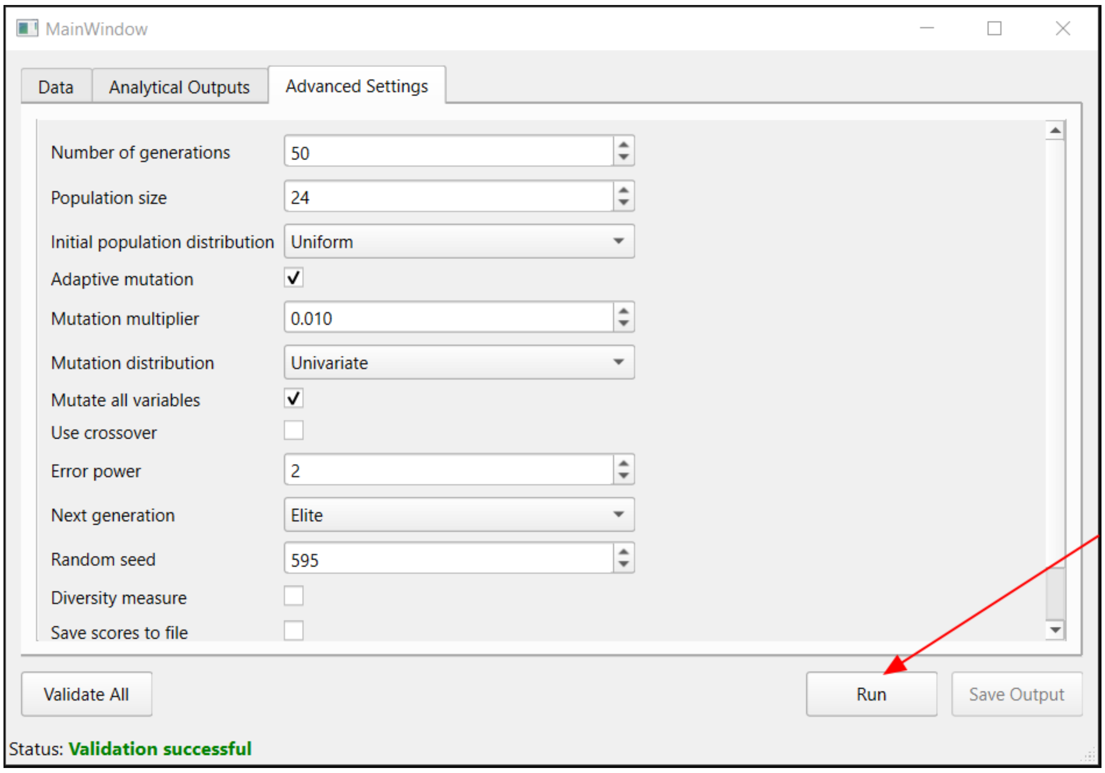
Data | Analytical Outputs | Advanced Settings

Number of generations: 50

Population size: 24

Initial population distribution: Uniform

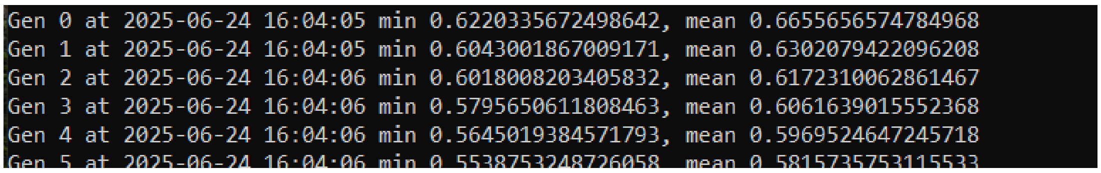
7. Click on the “Run” button, on the bottom right, to run the EA:



Number of generations 50
Population size 24
Initial population distribution Uniform
Adaptive mutation ✓
Mutation multiplier 0.010
Mutation distribution Univariate
Mutate all variables ✓
Use crossover
Error power 2
Next generation Elite
Random seed 595
Diversity measure
Save scores to file

Status: **Validation successful**

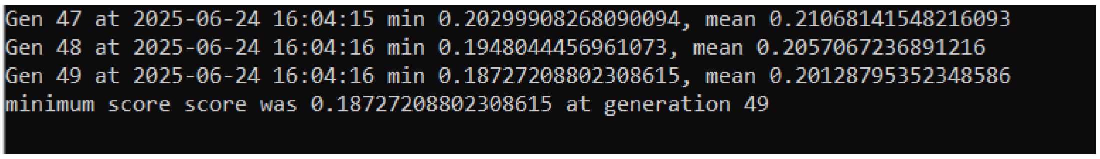
8. Look at the black console box and the output from the EA will be displayed:



```
Gen 0 at 2025-06-24 16:04:05 min 0.6220335672498642, mean 0.6655656574784968
Gen 1 at 2025-06-24 16:04:05 min 0.6043001867009171, mean 0.6302079422096208
Gen 2 at 2025-06-24 16:04:06 min 0.6018008203405832, mean 0.6172310062861467
Gen 3 at 2025-06-24 16:04:06 min 0.5795650611808463, mean 0.6061639015552368
Gen 4 at 2025-06-24 16:04:06 min 0.5645019384571793, mean 0.5969524647245718
Gen 5 at 2025-06-24 16:04:06 min 0.5538753248726058, mean 0.5815735753115533
```

This shows the generation number (e.g., “Gen 0”), the date and time, the minimum (optimal) fitness score for that generation and the mean fitness score for that generation.

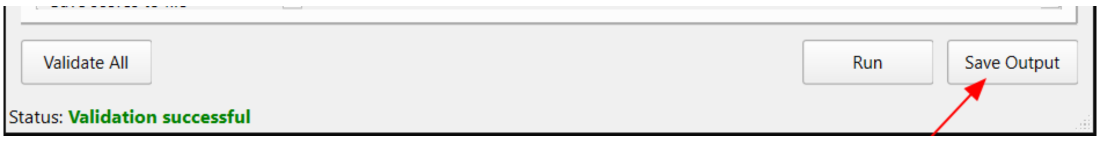
9. Once the EA has finished (in this example it will have run for 50 generations, labelled from 0 to 49), then a sentence is printed to summarize the run:



```
Gen 47 at 2025-06-24 16:04:15 min 0.20299908268090094, mean 0.21068141548216093
Gen 48 at 2025-06-24 16:04:16 min 0.1948044456961073, mean 0.2057067236891216
Gen 49 at 2025-06-24 16:04:16 min 0.18727208802308615, mean 0.20128795352348586
minimum score score was 0.18727208802308615 at generation 49
```

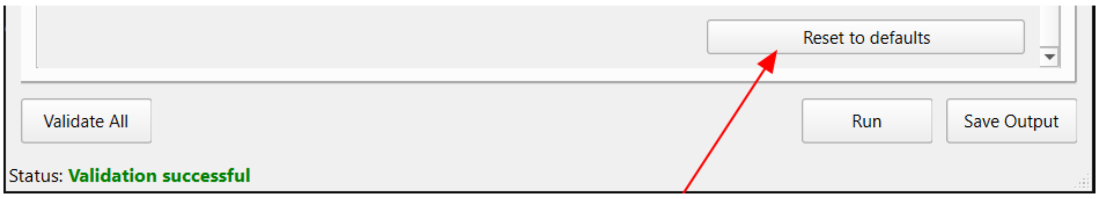
This shows that the minimum score was obtained on the last generation. (This indicates that 50 generations was probably not enough and we should run the EA for more generations than this, but it was fine for the purpose of this demonstration). We want the score to be as close to zero as possible.

10. The final step is to save the synthetic dataset by clicking on “Save output”, where a save file box will appear and you can navigate to where you would like to save the csv file of the optimal synthetic dataset:



Status: Validation successful

11. If you would like to perform multiple runs simply change the random seed (and any other parameters) and click on “Run” (you do not need to reload all the csv files). You can reset the parameters to their default settings at any time using the “Reset to defaults” button:

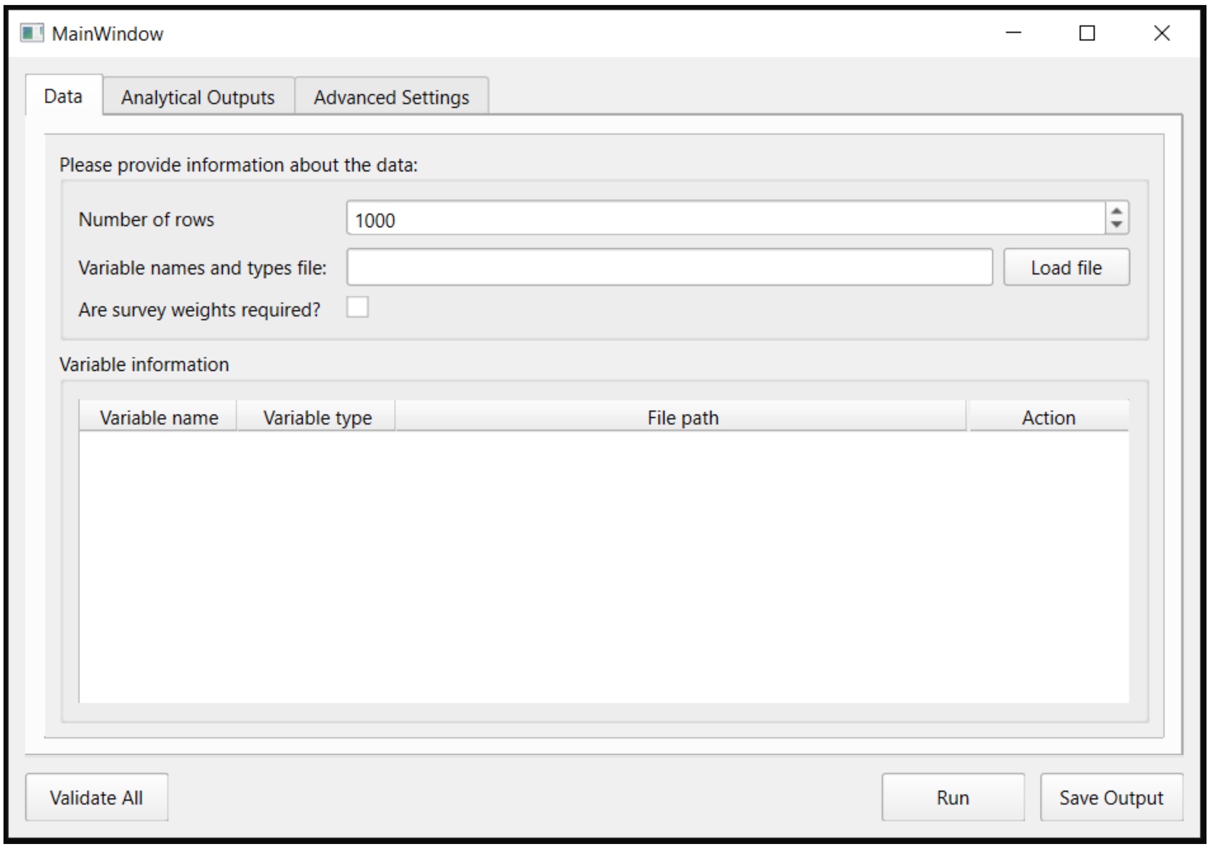


Status: Validation successful

4 Key Features and Functions

This section provides a more in depth look at the GUI and explains the Data and Analytical Outputs tabs.

4.1 Data Tab



Please provide information about the data:

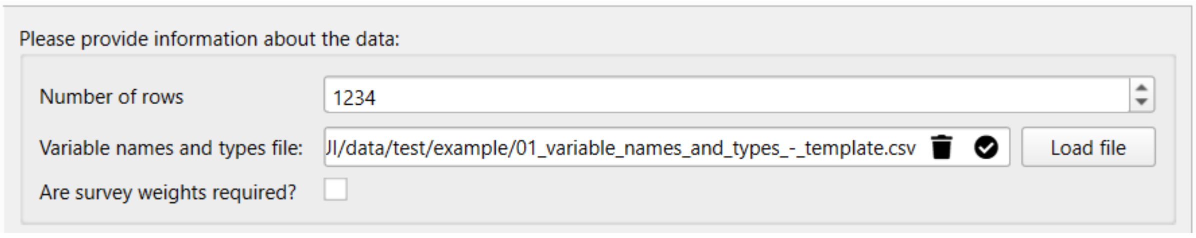
Number of rows	1000
Variable names and types file:	
Are survey weights required?	☐

Variable information

Variable name	Variable type	File path	Action

The Data tab has two sections. The first section collects information about the dataset (e.g. number of rows required), and the second section (Variable information) collects information about the individual variables in the dataset.

4.1.1 Information about the data

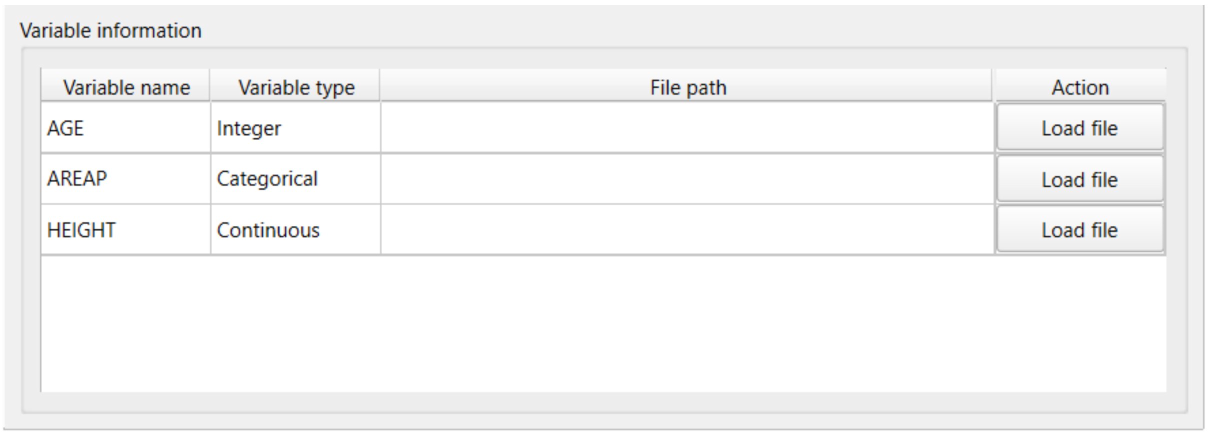

Please provide information about the data:

Number of rows	1234
Variable names and types file:	JI/data/test/example/01_variable_names_and_types_-_template.csv
Are survey weights required?	☐

In the Data tab information should be provided about the data. Firstly, enter the number of rows (records) required for the synthetic dataset. Then the variable names and types file (an example of this is contained in the Template Guide in section 6) should be loaded by clicking the “Load file” button; this informs the app of the name and type of all variables to be synthesised. Ticking the survey weights checkbox adds a column of survey weights to the synthetic dataset (use only if required).

The loaded file can be removed by clicking the trash icon. The validation status of a loaded file is shown with a circle icon (see section 4.3.1 for more information).

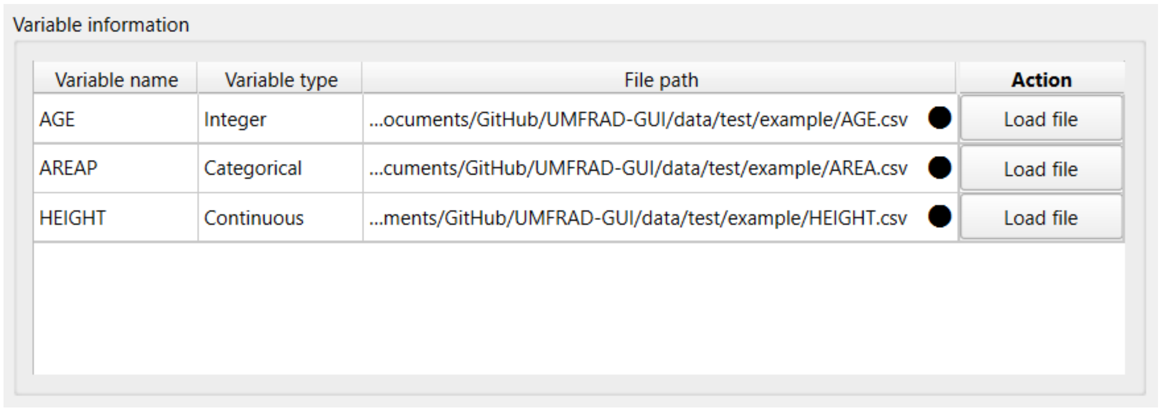
4.1.2 Variable information


Variable information

Variable name	Variable type	File path	Action
AGE	Integer		Load file
AREAP	Categorical		Load file
HEIGHT	Continuous		Load file

The table in the “Variable information” section will be loaded with information from the “variable names and types” file (if it is correctly formatted, see Template Guide in section 6). This details the variable name and type for each variable to be synthesised.

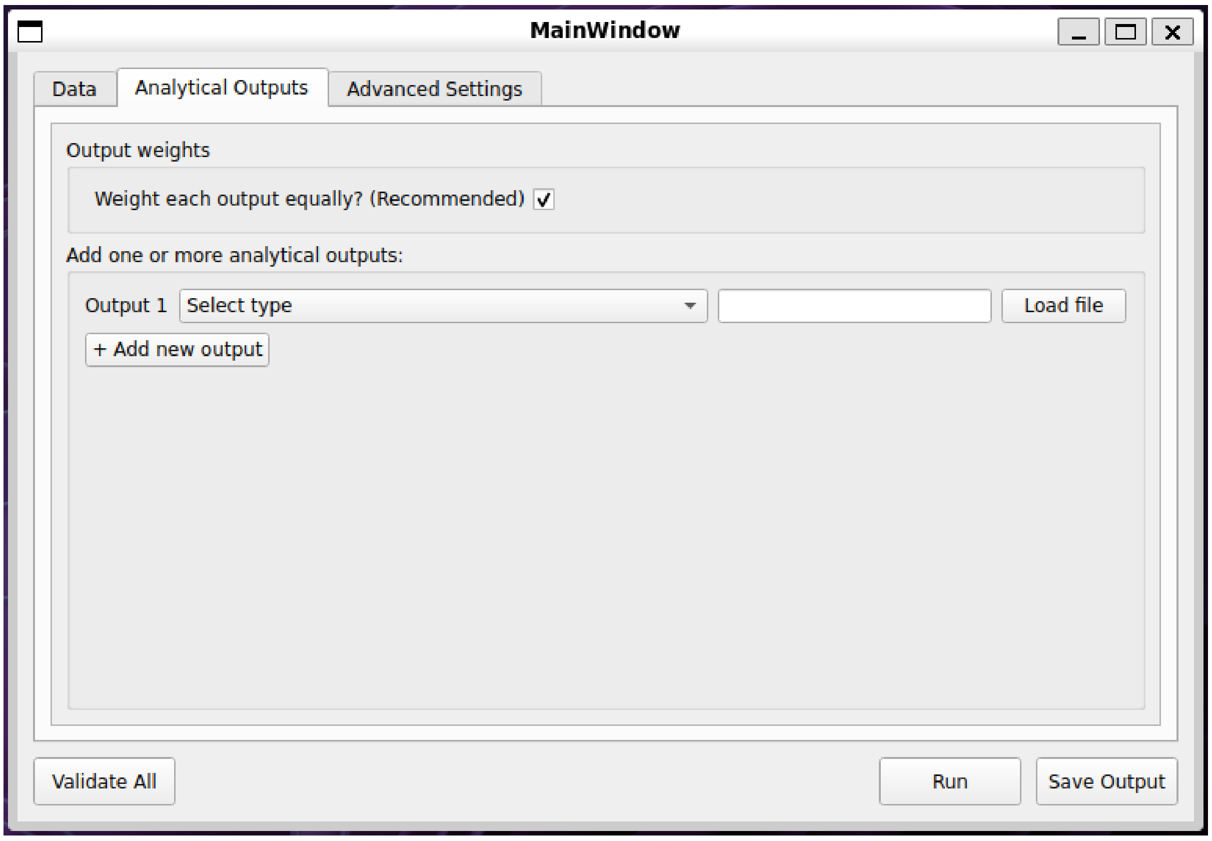
For each of the variables a data file (in the format of templates 02 and 03, see Template Guide section 6.2) should be loaded by clicking the “Load file” button in the corresponding row. This provides information to the app about the distribution of each variable.


Variable information

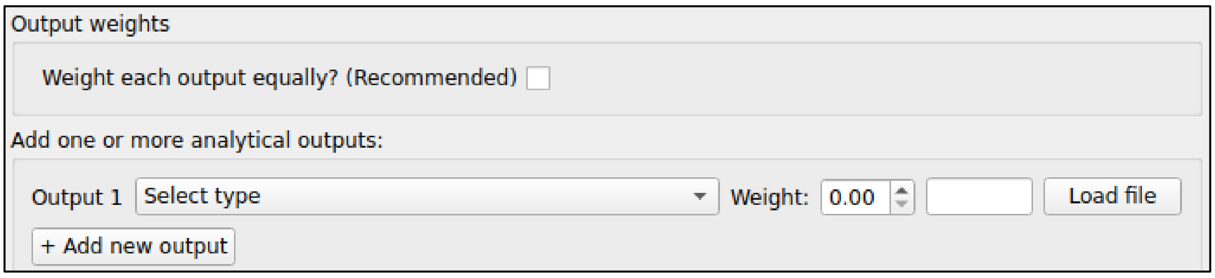
Variable name	Variable type	File path	Action
AGE	Integer	...ocuments/GitHub/UMFRAD-GUI/data/test/example/AGE.csv ●	Load file
AREAP	Categorical	...cuments/GitHub/UMFRAD-GUI/data/test/example/AREA.csv ●	Load file
HEIGHT	Continuous	...ments/GitHub/UMFRAD-GUI/data/test/example/HEIGHT.csv ●	Load file

4.2 Analytical Outputs tab

The Analytical Outputs tab collects the information for analytical outputs of the data and controls how to weight them (if there is more than one).



4.2.1 Output weights

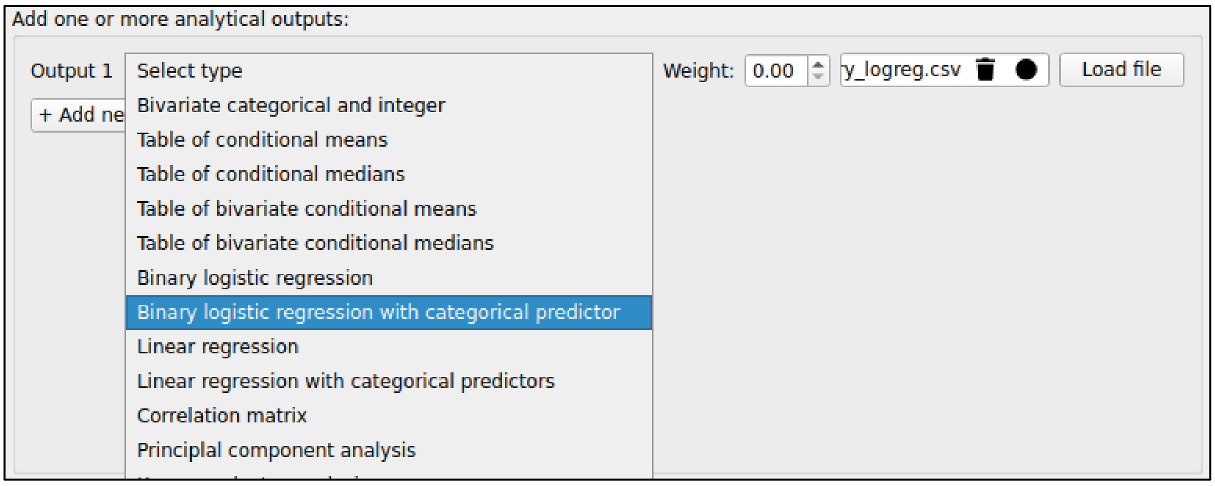

Output weights
Weight each output equally? (Recommended)
Add one or more analytical outputs:
Output 1
Select type
Weight: 0.00
Load file
+ Add new output

If more than one analytical output is loaded, the user can choose how they are weighted. For example, if two outputs are loaded and one is deemed more important than the other, then this one can be weighted higher and so it will assume more importance. By default, the outputs are weighted equally, and this is the recommended setting.

To manually assign a weight to each analytical output, untick the “Weight each output equally? (Recommended)” checkbox. Each analytical output will then have a weight field. This has a default value of 0, takes a minimum value of 0 and a maximum value of 1. The weights of all analytical outputs should add up to 1.

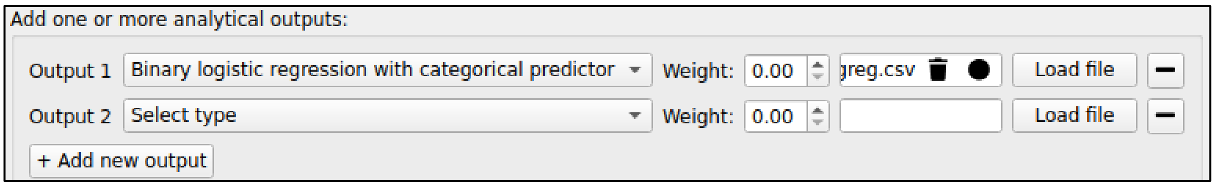
4.2.2 Analytical outputs

At least one analytical output file must be loaded – this is what drives the EA and the aim of the synthetic data is that it will reproduce that same analytical output when analysed.


Add one or more analytical outputs:
Output 1
+ Add ne
Select type
Bivariate categorical and integer
Table of conditional means
Table of conditional medians
Table of bivariate conditional means
Table of bivariate conditional medians
Binary logistic regression
Binary logistic regression with categorical predictor
Linear regression
Linear regression with categorical predictors
Correlation matrix
Principal component analysis
Weight: 0.00
y_logreg.csv
Load file

For each analytical output a csv file must be loaded (see the Template Guide in section 6.3 for further information) to describe it:

- Click on the dropdown menu and select the type of analytical output from the available options
- Enter a weight (if required, see section 4.2.1)
- Load the analytical output file by clicking the "Load file" button



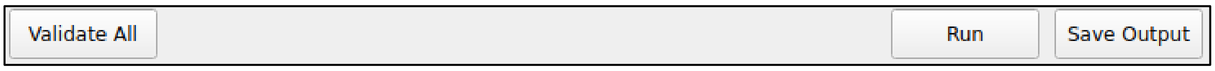
A new analytical output can be added by clicking the "+ Add new output" button.

If an analytical output is no longer required, it can be removed by clicking the "-" button on the right of each analytical output row. The delete button is only shown when there are two or more analytical outputs.

There must be at least one analytical output.

4.3 Main Buttons

The main buttons of the GUI are the "Validate All", "Run" and "Save Output" buttons.



4.3.1 Validation

The inputs and the loaded files will be checked (or validated) when the "Validate All" button is clicked. A dialog box will appear once the validation is complete, either confirming that all inputs and files are valid, or highlighting any invalid inputs or files.

Icon	Validation status
●	Pending
◉	In progress
✔	Valid
✖	Invalid

4.3.2 Running the app

The inputs and data from the loaded files will be used to run the EA when the “Run” button is clicked.

If the “Save scores to file” setting (see section 5.1.2) is selected, a file dialog box will pop up for you to choose a location to save the fitness scores. This is specified before the EA is run.

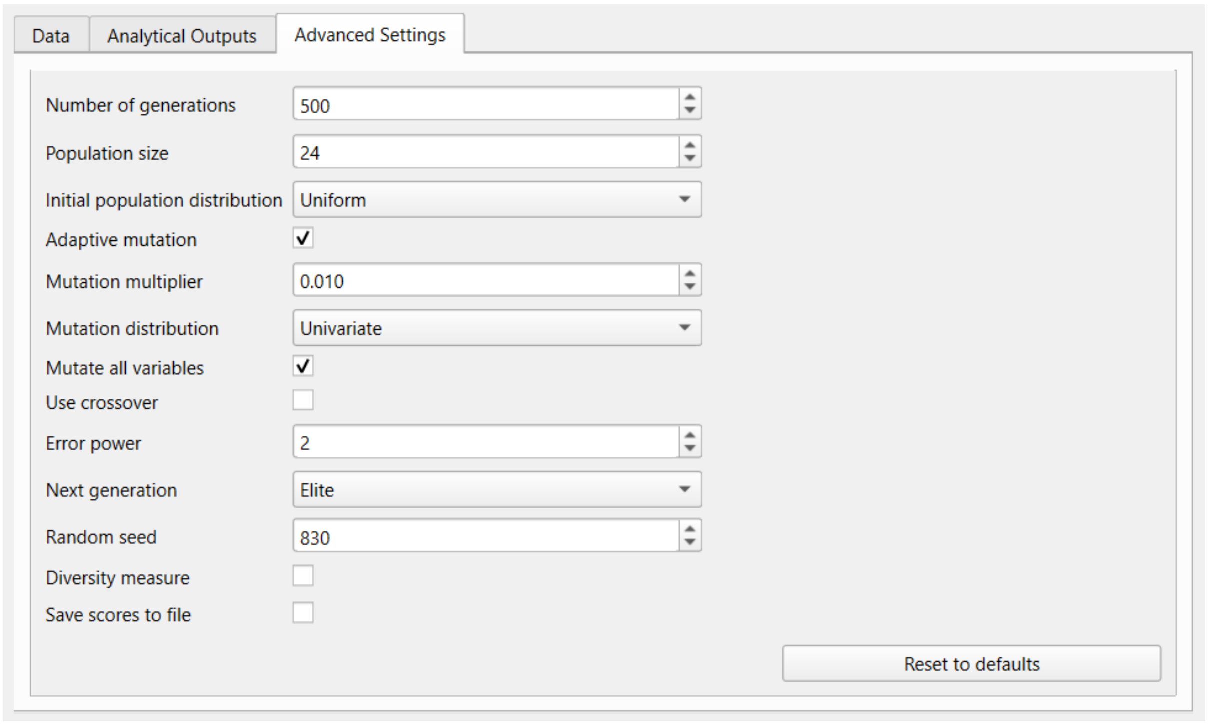
4.3.3 Saving output

Once the app stops running, to save the generated synthetic dataset click the “Save Output” button.

5 Advanced Features

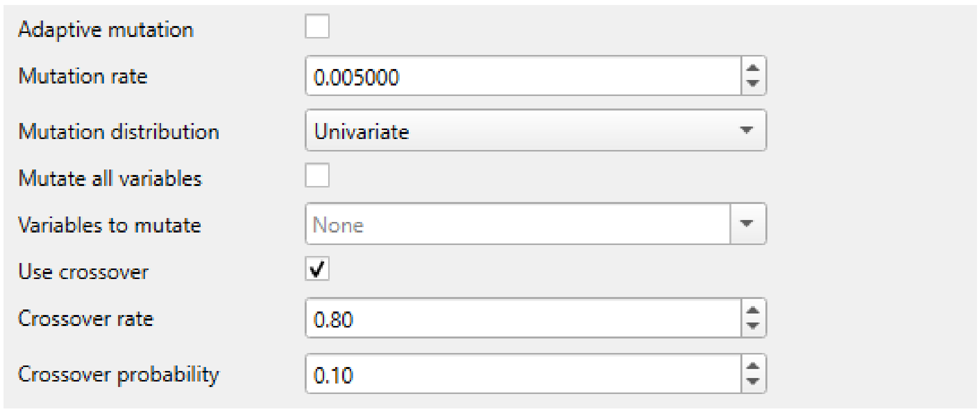
There are various advanced settings that can be modified in order to (ideally) improve the performance of the app. This section explains the Advanced Settings tab.

5.1 Advanced Settings tab



Number of generations	500
Population size	24
Initial population distribution	Uniform
Adaptive mutation	✓
Mutation multiplier	0.010
Mutation distribution	Univariate
Mutate all variables	✓
Use crossover	
Error power	2
Next generation	Elite
Random seed	830
Diversity measure	
Save scores to file	

5.1.1 Interactive checkboxes



Adaptive mutation	☐
Mutation rate	0.005000
Mutation distribution	Univariate
Mutate all variables	☐
Variables to mutate	None
Use crossover	✓
Crossover rate	0.80
Crossover probability	0.10

The checkboxes for "Adaptive mutation", "Mutate all variables" and "Use crossover" are interactive checkboxes which affect the state (enabled/disabled, visible/invisible) of other input fields.

- Unticking the "Adaptive mutation" checkbox enables the "Mutation rate" input field.
- Unticking the "Mutate all variables" checkbox enables the "Variables to mutate" input field, where variables can be chosen individually for mutation (rather than using all)
- Ticking the "Use crossover" checkbox shows the "Crossover rate" and "Crossover probability" input fields.

5.1.2 Available settings

The following settings, or parameters, can be modified. Further information about these parameters can be found in Section 7.2.

- **Number of generations**: Enter the maximum number of generations required. The algorithm will stop once it reaches the specified number of generations.
  - Input constraints: Select an integer value between 1 and 100,000
- **Population size**: Enter the population size (number of synthetic datasets in each generation) for the algorithm.
  - Input constraints: Select an even integer value between 2 and 100
- **Initial population distribution**: Select the initial population distribution.
  - Options: Uniform and Univariate
- **Adaptive mutation**: Enable or disable adaptive mutation for the algorithm.
- **Mutation multiplier**: When adaptive mutation is selected this value is multiplied by the fitness score to calculate the mutation rate.
  - Input constraints: Select a decimal between 0 and 1 up to 3 decimal places
- **Mutation rate**: When adaptive mutation is not selected, enter the mutation rate for the algorithm manually.
  - Input constraints: Select a decimal number with up to 6 decimal places between 0 and 1
- **Mutation distribution**: Select the mutation distribution for the algorithm.
  - Options: Univariate and Uniform
- **Mutate all variables**: Enable or disable the mutation of all variables
- **Variables to mutate**: Select any number of variables to mutate.
  - Options: A list of variables already loaded in the Data tab (see section 4.1)

- **Use crossover**: Enable or disable the use of crossover. Show or hide crossover rate and crossover probability settings.
- **Crossover rate**: Enter the crossover rate.
  - Input constraints: Select a decimal number with up to 2 decimal places between 0 and 1
- **Crossover probability**: Enter the crossover probability.
  - Input constraints: Select a decimal number with up to 2 decimal places between 0 and 1
- **Error power**: Enter the error power, the default value of 2 corresponds to using Mean Squared Error (MSE).
  - Input constraints: Select an integer value between 1 and 10
- **Next generation**: Choose the selection strategy for the next generation.
  - Options: Elite and Children
- **Random seed**: Enter a number to be used as the random seed in the algorithm.
  - Input constraints: Select an integer value between 0 and 1000
- **Diversity measure**: Enable or disable the diversity function.
- **Save scores to file**: Enable or disable saving the fitness scores to a csv file.
  - Fitness scores are saved to file when this option is checked
  - Diversity and combined scores are also saved to files when the 'Diversity measure' option is checked in addition to this option

5.1.3 Restoring the default settings

Clicking the "Reset to defaults" button restores the advanced settings to their original values and generates a new random seed.

6 Template Guide

6.1 Introduction

In order to use the GUI, information needs to be loaded via various .csv files. Template files are provided to aid with this. This section describes the .csv template files and how to use them.

The template files are included with the installation files (in the "templates" folder), and they are numbered to allow easier identification. Cells highlighted in red within the files should not be changed (although as they are .csv files, the red highlights are only visible within this document).

6.2 Data templates

The following templates are used in the Data tab, to provide distributional information about the variables to be synthesised.

The process goes as follows:

- First, input the variable names and types (using template 01)
- Secondly, input the information for each of the variables (one csv file for each variable)
  - Categorical or integer variables use template 02
  - Continuous variables use template 03

6.2.1 Template 01 – Variable names and types

Filename: “01_variable_names_and_types.csv”

This is a **required** file and provides information to the EA about the name and data type of each variable to be synthesised. The example below shows three variables (AGE, AREA, HEIGHT), each of a different type:

Variable_name	Type
AGE	Integer
AREA	Categorical
HEIGHT	Continuous

“Variable_name” and “Type” (highlighted in red) are the headers for columns 1 and 2 respectively, **these should not be changed**.

- First column contains the variable name
- Second column must contain either “Integer”, “Categorical” or “Continuous” (cannot be blank)

6.2.2 Template 02 – Categorical and integer variables

Filename: “02_univariate_categorical_and_integer_-_template.csv”

For each categorical or integer variable a .csv file is required to describe it, using this template (do not delete cells in red). An example is detailed below, using the variable “AREA”:

Input Type	univ_tab	
AREA	Proportion	Missing
1	0.9775	
2	0.0125	
-99	0.01	X

Column 1, below “Input_type” should contain **a valid variable name** (from file 01) to replace the example “AREA”, and below this you should input integers representing the values that the variable can take.

Column 2, below “Proportion” should contain the corresponding **proportions** that each value occurs. If the proportions are not known, then the easiest option is to set them all equal (e.g. if there are two values, set them both to 0.5).

Column 3, below “Missing” - if there are values that you would like to be treated as missing (e.g. those representing empty cells or values that you do not want to be included in calculations such as means or regressions) put an “X” in the corresponding column.

6.2.3 Template 03 – continuous variables

Filename: “03_univariate_continuous_-_template.csv”

For each continuous variable a .csv file is required to describe it, using this template. An example is detailed below, using the variable “PAY”:

Input Type	cont_var	
PAY	Proportion	Missing
-99	0.01	X
Max	100000	
Min	0	
Mean	30153.25	
Standard deviation	7002.37	
Skew		
Distribution	Normal	

Column 1, below "Input Type" should contain **a valid variable name** (from file 01) to replace "PAY". Below this if there are any values to be treated as missing, they should be entered here – in this example -99 is a missing value. **This row should be deleted if there are no missing values**.

Column 2, below "Proportion", if there are any missing values enter the corresponding proportion that they occur (this can be an estimate if unknown). Then corresponding to the variables in column 1 (Max, Min, etc.) the following values should be entered:

- "Max" – maximum value the variable can take (**required**, use an estimate if unknown)
- "Min" – minimum value the variable can take (**required**, use an estimate if unknown)
- "Mean" – mean value (if known, otherwise leave blank)
- "Standard deviation" – standard deviation (if known, otherwise leave blank)
- "Skew" – skew (if known, otherwise leave blank)
- "Distribution" – distribution can be "Normal", "Poisson" or blank

Column 3, below "Missing" – if there are any missing values listed, put an "X" in the corresponding column.

6.3 Analytical output templates

The second tab of the GUI is where the analytical outputs are loaded. At least one analytical output must be provided. Fourteen different types of analytical output can be used, these are:

- Univariate table
- Univariate continuous variable
- Bivariate table
- Table of conditional means
- Table of conditional medians
- Table of bivariate conditional means
- Table of bivariate conditional medians
- Binary logistic regression
- Binary logistic regression with categorical predictors
- Linear regression
- Linear regression with categorical predictors
- Correlation matrix
- Principal component analysis
- K-means cluster analysis

Each individual output requires a .csv template to load the information, these are described below:

6.3.1 Template 02 – Univariate categorical or integer table

Filename: "02_univariate_categorical_and_integer_-_template.csv"

Analytical output: a table detailing the values and proportions of a single variable.

This file is used to input the information about the variable (see Data section 6.2), but it can also to be used to represent an analytical output if required (and is filled out in exactly the same way).

6.3.2 Template 03 – Univariate continuous

Filename: "03_univariate_continuous_-_template.csv"

Analytical output: a single continuous variable, with mean (and optionally standard deviation and skew) included.

This file is used to input the information about the variable (see Data 6.2 section), but it can also to be used to represent an analytical output if required (and is filled out in exactly the same way).

6.3.3 Template 04 – Table: bivariate categorical or integer

Filename: "04_bivariate_categorical_and_integer_-_template.csv"

Analytical output: a bivariate (two-way) table.

Input Type	biv_tab	
	SEX	
AREA	1	2
1	0.4	0.34
2	0.14	0.1
-9	0.01	0.01

In the above example, the first row (in red) should not be changed, and below this is a two-way table with the values and proportions for AREA crossed with SEX. The table can be of any size.

In Column 1, "AREA" should be replaced by a valid variable name, and below this should be the values it can take.

In Column 2, "SEX" should be replaced by a valid variable name and the **row** below this contains the values it can take.

The proportions should be inputted in the correct place in the table to correspond to the given values. Do not leave blank cells, input zero instead.

6.3.4 Template 05 – Table of conditional means

Filenames: "05_table_of_conditional_means_-_template.csv"

Analytical output: a table of conditional means. For example, the below table details the mean of PAY for each category of AREA.

Input Type	univ_mean_tab
AREA	PAY
1	30012.35
2	20005.76
-99	

In the above example, the top row (in red) should remain unchanged.

Column 1, below "Input Type" a valid variable name should be inserted, below this should be the values it can take.

Column 2, below "univ_mean_tab" a valid variable name should be inserted, and below this are the mean values.

In this example, a value (-99) is listed, and no mean is provided – leave the corresponding second column blank if you do not require a mean to be calculated for a value. In this example -99 would represent a missing value.

6.3.5 Template 06 – Table of conditional medians

Filenames: "06_table_of_conditional_medians_-_template.csv"

Analytical output: a table of conditional medians. For example, the below table details the median of PAY for each category of AREA.

Input Type	univ_med_tab
AREA	PAY
1	27454.45
2	18786.34
-99	

In the above example, the top row should remain unchanged.

Column 1, below "Input Type" a valid variable name should be inserted, below this are the values it can take.

Column 2, below "univ_med_tab" a valid variable name should be inserted, and below this are the median values.

In this example, a value (-99) is listed, and no median is provided – leave the corresponding second column blank if you do not require a median to be calculated for a value. In this example -99 would represent a missing value.

6.3.6 Template 07 – Table of bivariate conditional means

Filename: "07_table_of_bivariate_conditional_means_-_template.csv"

Analytical output: a table of bivariate (two-way) conditional means. For example, the below table details the mean of PAY for the various categories of AREA by SEX.

Input Type	biv_mean_tab	
Means of	PAY	

	SEX	
AREA	1	2
1	20200.23	20203.45
2	16436.45	19001.67
-99		

In the above example cells highlighted in red should remain unchanged. The example contains a table of AREA by SEX where each value is the mean of PAY for that combination. The table can be of any size.

Column 1, replace "AREA" with a valid variable name, then list below this the values it can take.

Column 2, replace "PAY" with the variable from which the means are calculated, below this replace "SEX" with the other variable from the bivariate table. In the **row** below this list the variable values.

The mean values are inserted at the required position in the table.

In this example a value (-99) is listed, and no mean is provided – leave the corresponding columns blank if you do not require a mean to be calculated. In this example -99 would represent a missing value.

6.3.7 Template 08 – Table of bivariate conditional medians

Filename: "08_table_of_bivariate_conditional_medians_-_template.csv"

Analytical output: a table of bivariate (two-way) conditional medians. For example, the below table details the median of PAY for the various categories of AREA by SEX.

Input Type	biv_med_tab	
Means of	PAY	
	SEX	
AREA	1	2
1	20200.23	20203.45
2	16436.45	19001.67
-99		

In the above example cells highlighted in red should remain unchanged. The example contains a table of AREA by SEX where each value is the median of PAY for that combination. The table can be of any size.

Column 1, replace "AREA" with a valid variable name, then list below this the values it can take.

Column 2, replace "PAY" with the variable from which the medians are calculated, below this replace "SEX" with the other variable from the bivariate table. In the **row** below this list the variable values.

The median values are inserted at the required position in the table.

In this example a value (-99) is listed, and no median is provided – leave the corresponding columns blank if you do not require a median to be calculated. In this example -99 would represent a missing value.

6.3.8 Template 09 – Binary logistic regression

Filename: “09_binary_logistic_regression_-_template.csv”

Analytical output: a binary logistic regression without categorical predictors (i.e. there is no need for reference levels).

Input Type	log_reg	
Target	CARS	
Log-Likelihood	553.577	
Variables	B	S.E.
Constant	6.435	0.023
AGE	-0.008	0.091
PAY	-0.008	0.006
HEIGHT	-0.022	0.015

In the above example, cells highlighted in red should not be changed. This template contains the “Target” variable (**required**), “Log-likelihood” value (**optional**) and a table of regression coefficients (**required**) and standard errors (**optional**).

Column 1, below “Constant” include the variable names of the predictors.

Column 2, below “log_reg” include the variable name of the target variable, (in this example it is CARS). Below this the log-likelihood value can be included (leave blank if not known or not required). Below the “B” the constant can be specified if known, leave blank otherwise. Below this the regression coefficients can be included (**required**) to correspond with the variable names.

Column 3, below “S.E.” the Standard Errors can be included (leave blank if not known or not required)

6.3.9 Template 10 – Binary logistic regression with categorical predictors

Filename: “10_binary_logistic_regression_with_categorical_predictors_-_template.csv”

Analytical output: a binary logistic regression with at least one categorical predictor (i.e. reference levels are required).

Input Type	log_reg_with_cat	
Target	CARS	
Log-Likelihood	553.577	
Variables	B	S.E.
Constant	1.019	0.538
AGE	-0.012	0.01
PAY	-0.01	0.007
HEIGHT	-0.013	0.015
TENURE_1_ref		
TENURE_2	-0.436	0.381
TENURE_3	-0.632	0.341
TENURE_4	-0.603	0.318

TENURE_5	-1.267	0.447
TENURE_6	-1.5	0.835
TENURE_7	-1.231	0.391

In the above example, cells highlighted in red should not be changed. This template contains the “Target” variable (**required**), “Log-likelihood” value (**optional**) and a table of regression coefficients (**required**) and standard errors (**optional**).

Column 1, below “Constant” include the variable names of the predictors. For categorical variables include each category in the format “variablename_category”, with the reference level indicated by adding “_ref” to the end. In the above example TENURE is categorical, with seven categories, and category 1 is the reference level.

Column 2, below “log_reg_with_cat” include the variable name of the target variable, (in this example it is CARS). Below this the log-likelihood value can be included (leave blank if not known or not required). Below the “B” the constant can be specified if known, leave blank otherwise. Below this the regression coefficients can be included (**required**) to correspond with the variable names and category levels – leave the reference levels blank.

Column 3, below “S.E.” the Standard Errors can be included (leave blank if not known or not required)

6.3.10 Template 11 – Linear regression

Filename: “11_linear_regression_-_template.csv”

Analytical output: a linear regression without categorical predictors (i.e. there is no need for reference levels).

Input Type	lin_reg	
Target	PAY	
R-squared	0.35	
Variables	B	S.E.
Constant	0.249	0.425
WEIGHT	-0.008	0.006
HEIGHT	-0.022	0.015

In the above example, cells highlighted in red should not be changed. This template contains the “Target” variable (**required**), “R-squared” value (**optional**) and a table of regression coefficients (**required**) and standard errors (**optional**).

Column 1, below “Constant” include the variable names of the predictors.

Column 2, below “lin_reg” include the variable name of the target variable, (in this example it is PAY). Below this the R-squared value can be included (leave blank if not known or not required). Below the “B” the constant can be specified if known, leave blank otherwise. Below this the regression coefficients can be included (**required**) to correspond with the variable names.

Column 3, below “S.E.” the Standard Errors can be included (leave blank if not known or not required)

6.3.11 Template 12 – Linear regression with categorical predictors

Filename: “12_linear_regression_with_categorical_predictors_-_template.csv”

Analytical output: a linear regression with at least one categorical predictor (i.e. reference levels are required).

Input Type	lin_reg_with_cat	
Target	WEIGHT	
R-squared	0.35	
Variables	B	S.E.
Constant	1.019	0.538
AGE	-0.012	0.01
PAY	-0.01	0.007
HEIGHT	-0.013	0.015
TENURE_1_ref		
TENURE_2	-0.436	0.381
TENURE_3	-0.632	0.341
TENURE_4	-0.603	0.318
TENURE_5	-1.267	0.447
TENURE_6	-1.5	0.835
TENURE_7	-1.231	0.391

In the above example, cells highlighted in red should not be changed. This template contains the “Target” variable (**required**), “R-squared” value (**optional**) and a table of regression coefficients (**required**) and standard errors (**optional**).

Column 1, below “Constant” include the variable names of the predictors. For categorical variables include each category in the format “variablename_category”, with the reference level indicated by adding “_ref” to the end. In the above example TENURE is categorical, with seven categories, and category 1 is the reference level.

Column 2, below “lin_reg_with_cat” include the variable name of the target variable, (in this example it is WEIGHT). Below this the R-squared value can be included (leave blank if not known or not required). Below the “B” the constant can be specified if known, leave blank otherwise. Below this the regression coefficients can be included (**required**) to correspond with the variable names and category levels – leave the reference levels blank.

Column 3, below “S.E.” the Standard Errors can be included (leave blank if not known or not required)

6.3.12 Template 13 – Correlation matrix

Filename: “13_correlation_matrix _-_template.csv”

Analytical output: a correlation matrix.

Input Type	**corr_mat**		
Correlation	Pearson		
	AGE	PAY	HEIGHT
AGE	1	0.23	0.07

PAY	0.23	1	0.02
HEIGHT	0.07	0.02	1

In the above example, cells highlighted in red should not be changed. This template contains the type of correlation (**required**) and then a table with the correlation values (**required**).

Row 1, in the cell next "Correlation" include the correlation type – either "Pearson", "Kendall" or "Spearman".

Row 2 onwards, include correlation table, detailing the variable names (should be in the same order on both column and row) and the correlation values.

6.3.13 Template 14 – Principal component analysis

Filename: "14_principal_component_analysis_-_template.csv"

Analytical output: a principal component analysis

Input Type	pca	
KMO	0.513	
Rotation	None	
Component	% of Variance	
1	63.166	
2	28.432	
Factor loadings	Component	
Variable	1	2
AGE	0.924	-0.137
PAY	0.918	0.174
HEIGHT	0.525	0.831
CARS	-0.744	0.63
Correlation between	1	2
1	1	0.345
2	0.345	1

In the above example, cells highlighted in red should not be changed.

The template is split into 3 sections, and a gap row is purposefully left between them:

- Section 1 contains the KMO value (**optional**), the Rotation (**optional**) and the % of Variance (**required**).
- Section 2, Factor loadings (**required**)
- Section 3, Correlation between factors (**optional**), note that **if this section is not required it should be deleted**.

Section 1, KMO value is optional (leave blank if not required). Rotation can be any of "varimax", "oblimax", "quartimax", "equamax", "geomin_ort", "promax", "oblimin", "quartimin", "geomin_obl", please **write "None" if no rotation is required** (as shown in the above example). For the % of

Variance table label the components sequentially starting at 1 (e.g. if there are two, that would be 1, 2).

Section 2, for the "Factor loadings" table list the variable names and the components numbered numerically from 1 upwards.

Section 3, for the "Correlation between" factors include the correlation table with the factors labelled from 1 upwards. **Delete this section if not required.**

6.3.14 Template 15 – K-means cluster analysis

Filename: "15_k_means_cluster_analysis_-_template.csv"

Analytical output: a K-means cluster analysis

Input Type	k_means	
Final cluster centres	Cluster	
Variable	1	2
AGE	13.7	39.6
PAY	4.4	9
HEIGHT	16	20.5
CARS	1	0.6
Number of cases in each cluster	Cluster	N
	1	392
	2	232

In the above example, cells highlighted in red should not be changed.

This template consists of two sections, both of which are **required**:

- The "Final cluster centres", with variable names listed in the first column, and the following columns listing the cluster centres for each cluster by variable. Note that the clusters should be labelled from 1 upwards.
- The "Number of cases in each cluster", with the "Cluster" column identifying the cluster number (labelled from 1 upwards) and the "N" column including the number of records in that cluster.

7 Troubleshooting

7.1 Introduction

This section contains information that should help you to get the best out of the EA. A description of the parameters is included to start with as it may be that some issues might be solved by a greater understanding of what these do.

7.2 Parameters

The following describes the parameters and the various settings, understanding these may help with troubleshooting and improving performance.

Parameter	Description
Number of generations	The EA runs for a settable number of generations. By default, this is set to 500 but every dataset is different, and some may require fewer generations, whereas others may require many more. The larger the number of generations the longer the EA will take to run.
Population size	How many synthetic datasets make up the population. This should be an even number and by default is set to 24. The larger the population the more diverse it may be (which is good) but larger populations will usually result in a longer run time. We have achieved good results using a population size of 32.
Initial population distribution	When the initial population of datasets is generated, either the uniform or univariate distribution can be used to generate it. Note, that if only the uniform distribution is known this will be used when univariate is selected. Uniform is used by default as this leads to a more diverse starting population.
Adaptive mutation	When selected the EA uses an adaptive mutation rate. The adaptive mutation rate is a factor of the fitness error value and therefore falls as the error does. Adaptive mutation has proven to be optimal in testing.
Mutation multiplier	Only applicable when adaptive mutation is selected – this is the value by which the fitness is multiplied to get the adaptive mutation rate. The default value is 0.01
Mutation rate	Only applicable when adaptive mutation is not selected – the EA then uses a constant mutation rate. The default value is 0.005
Mutation distribution	Selects whether the mutation uses the univariate distribution or the uniform distribution. Using the univariate tends to provide better results. If the univariate is selected but only the uniform distribution is known, then uniform will be used.
Mutate all variables	By default, the EA will mutate all variables, however it is possible to select the variables to be mutated.
Variables to mutate	Only applicable when "Mutate all variables" is not selected. This is a dropdown of the variable names and allows selection of the variables to be mutated.
Use crossover	Selects whether or not crossover is used. By default, crossover is off, however in testing good results have been obtained when using crossover.
Crossover rate	Only applicable when "Use crossover" is selected. This is the probability of two parent synthetic datasets crossing over. It is set to 0.8 by default.
Crossover probability	Only applicable when "Use crossover" is selected. If two datasets are chosen for crossover this is the probability of individual records crossing over. E.g. if set at 0.1 which is the default, about 10% of records would be swapped.
Error power	The error between the synthetic and original output is what drives the objective function (we want to minimise this). The default value for error power is 2, meaning that the mean squared error is calculated. In testing this generally had optimal results.
Next generation	Selects whether the next generation will be composed of children, or the best synthetic datasets from the population (elite). Elite is the default setting.

Random seed	This sets a random seed. If you use the same random seed for multiple runs, the same initial starting population will be used
Diversity measure	This is an experimental setting and if used will slow the EA's performance down. Use this only if you are getting poor results and have already tried adjusting the other parameters. This attempts to introduce more diversity into each population by forcing the algorithm to consider the distance each synthetic dataset is away from the optimal dataset, as well as the fitness.
Save scores to file	If checked this will save all the fitness scores at each generation.

7.3 Frequently Asked Questions

What is an EA? An EA is an Evolutionary Algorithm. EAs perform iterative optimisation and are inspired by the process of natural selection. EAs can be used to solve complex problems, in this case to generate a synthetic dataset where we do not have access to the original dataset. A population of candidate solutions (in this case, synthetic datasets) evolves over generations, with better solutions more likely to survive and reproduce. At the first generation the synthetic datasets will be random noise, but over time they will ideally evolve to be close to the original dataset (or at least to be able to reproduce the analytical output from the original).

What is an analytical output? An analytical output is any statistic or table or analysis that has been drawn from the original dataset. This could be a table of means, or the coefficients of a linear regression or anything that has been published as an output of that dataset. The Analytical Outputs section in Appendix A contains further information on the currently available output types that can be used with the EA.

What if I don't know the distribution of each variable? A guess is better than nothing. The EA cannot run without some basic information about the variables. For categorical or integer variables you can use the uniform distribution (i.e., divide 1 by the number of values and use that as the proportion, e.g. if there were four values the proportion for each would be 0.25). For continuous variables at the least a maximum and minimum value would need to be estimated, but the mean (and skew, standard deviation, etc.) can be left blank if unknown.

How do I represent missing values? In order for missing values to be included in the synthetic data the EA has to be told about them. They will need to be assigned a value, such as -99 (or something that does not clash with existing values), and there is space to include them in the categorical/integer (template 02) and continuous (template 03) templates.

How do I save the synthetic dataset? Once the EA has finished its run click on the Save Output button and a box will appear allowing you to choose where to save it.

I have received an error message, and the EA did not run. The first thing to do would be to check that all the templates are correctly formatted using the template guide. Check that variable names are correct and that the format of each template has not been altered. Another thing to check is the number of rows in the synthetic data – too few can lead to errors with some of the statistical functions – so try increasing the number of rows (if it is below 100)

The EA is taking too long to run. Try fewer generations, and/or a smaller population

The results are not what I expected, how do I improve performance? There are multiple settings in the Advanced Settings section, these can be finetuned to improve performance. Things to try include:

- Increase the number of generations – the EA may just need longer to train
- Perform multiple runs using different random seeds – some runs can be much better/worse than others
- Use crossover (or turn it off if already using it)
    - Within the crossover settings you can also experiment with the probability of crossover and how many records to crossover
- Try a constant mutation rate, such as 0.005

Appendix A – Analytical Outputs Information

Introduction

The EA requires at least one analytical output in order to produce the synthetic data; this section describes each of them. As the EA is coded in python it also describes which python package/s is used to process that output. In general, outputs in a table format are represented by a pandas dataframe, but for some of the more complex outputs there is a description of which packages are used to generate them.

Analytical outputs

There are 14 different types of analytical output, each of which is inputted using a .csv file (details of which are in the Template Guide, section 6). The following gives a brief description of each output type.

Table of univariate categorical or integer

This is a table that provides the distribution of a particular variable. For example:

Value	Proportion
1	0.9775
2	0.0125
-99	0.01

A table like this can be used to describe categorical or integer data. This type of table is also used as an input to describe the data (the information provided is used for mutation), so the synthetic dataset should maintain these patterns without using it as an analytical output. Nevertheless, this table might be used as an analytical output in order to be especially sure that the distribution is maintained in the synthetic dataset. Template 02 is used to upload this analytical output. The table is represented in a pandas dataframe.

Univariate mean

This analytical output is the mean of a continuous variable. Optionally the standard deviation and skew can also be included. Template 03 is used to upload this output. The numpy package is used to calculate the mean, standard deviation and skew values.

Table of bivariate categorical or integer

This is a two-way table that provides the distribution of two variables interaction. For example:

	SEX	
AREA	**1**	**2**
1	0.4	0.34
2	0.14	0.1
-9	0.01	0.01

Template 04 is used to upload this output. The table is represented in a pandas dataframe.

Table of conditional means

This is a table where the mean values of one variable are provided for each category of another variable. For example:

AREA	PAY
1	30012.35
2	20005.76
-99	

In this table you should include all values which the variable can take (in this example 1, 2 and -99 are included, where -99 represents a missing value), but if you do not require a mean to be calculated for a particular value then this cell should be left blank (e.g. the corresponding cell for the missing value, -99, is left blank). Template 05 is used to upload this analytical output. The table is represented in a pandas dataframe, and the means are calculated using numpy.

Table of conditional medians

This is a table where the median values of one variable are provided for each category of another variable. For example:

AREA	PAY
1	27454.45
2	18786.34
-99	

In this table you should include all values which the variable can take (in this example 1, 2 and -99 are included, where -99 represents a missing value), but if you do not require a median to be calculated for a particular value then this cell should be left blank (e.g. the corresponding cell for the missing value, -99, is left blank). Template 06 is used to upload this. The table is represented in a pandas dataframe, and the medians are calculated using numpy for unweighted calculations and wquantiles for weighted calculations.

Table of bivariate conditional means

This is a table of bivariate (two-way) conditional means. For example, in the following table the means of the PAY variable are given for the various combinations of AGE and SEX:

	SEX	
AREA	1	2
1	20200.23	20203.45
2	16436.45	19001.67
-99		

In this table you should include all values which the variable can take (in this example 1, 2 and -99 are included for AREA, where -99 represents a missing value), but if you do not require a mean to be calculated for a particular value then this cell should be left blank (e.g. the corresponding cells for the missing value, -99, are left blank)). Template 07 is used to upload this. The table is represented in a pandas dataframe, and the means are calculated using numpy.

Table of bivariate conditional medians

This is a table of bivariate (two-way) conditional medians. For example, in the following table the medians of the PAY variable are given for the various combinations of AGE and SEX:

	SEX	
AREA	1	2
1	20200.23	20203.45
2	16436.45	19001.67
-99		

In this table you should include all values which the variable can take (in this example 1, 2 and -99 are included for AREA, where -99 represents a missing value), but if you do not require a median to be calculated for a particular value then this cell should be left blank (e.g. the corresponding cells for the missing value, -99, are left blank). Template 08 is used to upload this. The table is represented in a pandas dataframe, and the medians are calculated using numpy for unweighted calculations and wquantiles for weighted calculations.

Binary Logistic regression (with/without categorical predictors)

This is a more complex output than the previous table-based outputs, but it will consist of a table of coefficients (column B in the example below) with the option of including standard errors (S.E in the example below). There is also an option to include the Log-likelihood value. See the Template Guide (section 6) for details of how to fill in the templates (template 09 without categorical predictors and template 10 with categorical predictors).

Note that the data is pre-processed before performing the logistic regression, and all rows with missing values (specified when inputting the information about each variable) are removed. Currently continuous variables are not standardised; this is an option that could be added in the future.

Variables	**B**	**S.E.**
Constant	6.435	
AGE	-0.008	
PAY	-0.008	
HEIGHT	-0.022	

The statsmodels package is used to perform the logistic regression; this was used because it has an option to include survey weights (if required).

Linear regression (with/without categorical predictors)

This is a more complex output than the previous table-based outputs, but it will consist of a table of coefficients (column B in the example below) with the option of including standard errors (S.E in the example below). There is also an option to include the R-squared value. See the template guide for details of how to fill in the templates (template 11 without categorical predictors and template 12 with categorical predictors).

Note that the data is pre-processed before performing the linear regression, and all rows with missing values (specified when inputting the information about each variable) are removed. Currently continuous variables are not standardised; this is an option that could be added in the future.

Variables	B	S.E.
Constant	0.249	
WEIGHT	-0.008	
HEIGHT	-0.022	

The statsmodels package is used to perform the linear regression; this was used because it has an option to include survey weights (if required).

Correlation matrix

A correlation matrix on a set of specified variables, such as:

	AGE	PAY	HEIGHT
AGE	1	0.23	0.07
PAY	0.23	1	0.02
HEIGHT	0.07	0.02	1

Three types of correlation can be calculated (Pearson, Spearman or Kendall). These are performed using the pandas package. If survey weights are included in the data there is also an option to calculate the weighted Pearson correlation; this uses the statsmodels package.

Note that the data is pre-processed before performing the correlation – all rows with missing values (specified when inputting the information about each variable) are removed.

Principal component analysis

A principal component analysis (PCA) on a set of specified variables. Required information for this is a table detailing the percentage of variance for each component, for example:

Component	% of Variance
1	63.166
2	28.432

and a table detailing the factor loadings, for example:

	Component	
Variable	1	2
AGE	0.924	-0.137
PAY	0.918	0.174
HEIGHT	0.525	0.831
CARS	-0.744	0.63

Optionally, the KMO value and correlation between the components can also be included. Rotation can also be specified (these can be any of None, varimax, oblimax, quartimax, equamax, geomin_ort, promax, oblimin, quartimin, geomin_obl). See the Template Guide (section 6) for details of how to fill in the template (template 14).

If correlation between the components is to be included this will only be calculated if the type of rotation is one of oblimin, quartimin, promax, geomin_obl, oblimax, otherwise there is no correlation.

The sklearn package performs the PCA, and the factor_analyzer package is used to calculate the KMO value and perform rotation.

Note that the data is pre-processed before performing the pca – all rows with missing values (specified when inputting the information about each variable) are removed, and the variables are standardised (using sklearn's StandardScaler).

It is important to note that we have discovered during testing **that sklearn implements PCA in a different way to SPSS**, so this may mean that the resulting synthetic data produces different results when tested using these methods. That is, if the synthetic data was optimised using the sklearn implementation and then it is analysed using SPSS the results may not appear to be optimal. An avenue for future work would be to offer different implementations of PCA to control for this.

There is no option to perform a weighted PCA – if survey weights are included in the synthetic data, they will be ignored.

K-means cluster analysis

A K-means cluster analysis on a set of specified variables. Required information for this is the final cluster centres, for example:

	Cluster	
Variable	1	2
AGE	13.7	39.6
PAY	4.4	9
HEIGHT	16	20.5
CARS	1	0.6

And the number of records in each cluster, for example:

Cluster	**N**
1	392
2	232

See the Template Guide (section 6) for details of how to fill in the template (template 15). K-means is implemented using sklearn.

Note that the data is pre-processed before performing the k-means analysis – all rows with missing values (specified when inputting the information about each variable) are removed, and the variables are standardised (using sklearn's StandardScaler).