Ref. Ares(2026)7680036 - 04/08/2026



Funded by the European Union

EuropeanCity2

EuropeanCity2	
Project number:	**101178170**
Project name:	European City Squared: Computational Social Science Simulation for Democracy
Topic:	HORIZON-CL2-2024-DEMOCRACY-01-06
Type of action:	HORIZON-RIA
Starting date of action:	01/01/2025
Project duration:	36 months
Project end date:	31/12/2027
Deliverable number:	D5.3
Deliverable title:	Generation of synthetic datasets
EC document version:	Ver1
WP number:	WP5
Lead beneficiary:	AU
Main author(s):	Kai Bäuerle, Yelena Guryanova
Internal reviewers:	Minsu Jiang, Luis Razo
Nature of deliverable:	DATA
Dissemination level:	PU
Delivery date from Annex 1:	M18
Actual delivery date:	M20

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Executive Agency (REA). Neither the European Union nor European Research Executive Agency (REA) can be held responsible for them.

Executive Summary

The aim of this task was utilise LLMs to generate sophisticated synthetic datasets, providing a realistic portrayal of individual voters preferences profiles which are grounded in existing data sources such as existing transcripts of deliberation sessions, social media data or discussion fora.

Early on in the scientific discussions it was established that LLMs are not the most appropriate tool for the job—this task is to do with faithfully reproducing correlations from a statistical sample of a voter population. It was decided to approach the problem mathematically and to train a network to be able to reproduce these patterns.

This code output by this deliverable generates synthetic survey respondents based on the behaviour of citizens of Aarhus, Denmark. The training was performed on anonymised data from a survey response questionnaire.

The core problem was that with only 500 useable respondents and 64 questions, there are far too many possible answer combinations to ever model the full picture directly. Instead, the system focuses on getting the pairwise relationships correct— e.g., how age relates to voting party, how education relates to opinion on an issue — for any demographic slice (age group, gender, etc.) one would care about.

The code delivers eight different generative approaches, implements and assesses them head-to-head — from simple statistical baselines to a GAN, a variational autoencoder, and a custom-built optimiser designed specifically for this task. Each is scored against a real-world ”noise floor” (how much even two genuine samples of real people naturally differ).

Key finding: the custom-built method produces the most statistically faithful synthetic population. Once a model is fit and saved, it can be reused to generate custom synthetic populations on demand (e.g., ”60 women aged 55–70”) for downstream research or reporting — without retraining and without touching the original respondent data again.

Contents

1 Description of task **3**

2 Z as a predictor of the probability distribution between X, Y **5**

3 Synthetic data generating machine **6**

4 Conclusion **8**

5 References **8**

1 Description of task

Members of the electorate in the city of Aarhus were asked to complete a survey on their opinions and voting preferences. The objective of this work is to use this data to create a machine that can synthesise individual voters faithfully, with a fidelity that corresponds closely to how they would answer the survey questions. To do this effectively, we must come up with a justified notion of 'distance' between the output of the machine and the information collected in the survey.

Our first observation is that survey data is *statistical*—we can make statements such as the fraction of voters who expressed a certain opinion, or the number of people who answered two or more questions in a particular direction. That is, our underlying aim is to create a machine that can create enough individuals, which are able to reproduce the statistics of the raw data. The data that we are working with has over $M = 60$ questions (features) and was collected from 500 voters, which we call the *electorate* $\mathcal{E}$. We shall use these round numbers to continue with the reasoning. The range of answers possible for each question, which we shall crudely take as 10 in order to produce an estimate of the size of the full correlation space—this has astronomically large dimension $d = 10^{60}$; if $x_i \in \mathcal{I}$ is the answer to question I, then the probability of a particular sequence of survey responses is

$$\{p(x_1, x_2, \cdots, x_M)\}, \quad p \in \mathbb{R}^{10^{60}}. \tag{1}$$

Let $\mathbf{x} = (x_1, x_2, \cdots, x_M)$ denote the outcome of a completed survey, as a lower bound, the minimum number of observations required to be able to begin to estimate p is 10 per outcome, which is 10^{61}. This also means that out of the 500 clean pieces of data we have at hand, the probability of any particular sequence $\mathbf{x}$ being observed is measure zero $\frac{500}{10^{60}} \sim 0$. So then where do the correlations live, and are there any that we can meaningfully observe or infer from the data? Out of the full correlation space which statistics is it relevant to reproduce faithfully and which of them are we prepared to forgo?

An important point to note is that typical measures of correlation, such as Pearson, Spearman and Kendall correlation will fail to capture important relationships in the data, because they all assume linearity or monotonicity between the variables. The Pearson correlation coefficient gives a measure of linearity between two variables (see Fig. 1); the Spearman and Kendall correlation coefficients both make the underlying assumption that the two variables are monotonically related and the two measures differ in nuanced statistical ways. *A priori* there is no reason to assume that any pair of variables in the survey data are monotonically related; indeed superficially one observes clustering and blob-like distributions.

1.1 Probability functions for pairs of random variables

To motivate our decision, imagine selecting two questions out of the survey. For this pair of questions, i.e., one can write down the set, or plot the associated discrete scatter plot, which

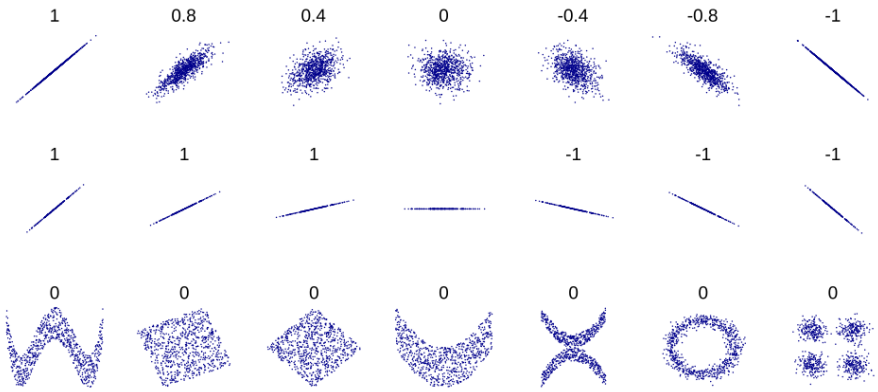


Figure 1: Discrete sets of points evaluated with the Pearson correlation coefficient for each set. The correlation reflects the noisiness and direction of a linear relationship (top row), but not the slope of that relationship (middle), nor many aspects of nonlinear relationships (bottom). Image from Wikipedea [1].

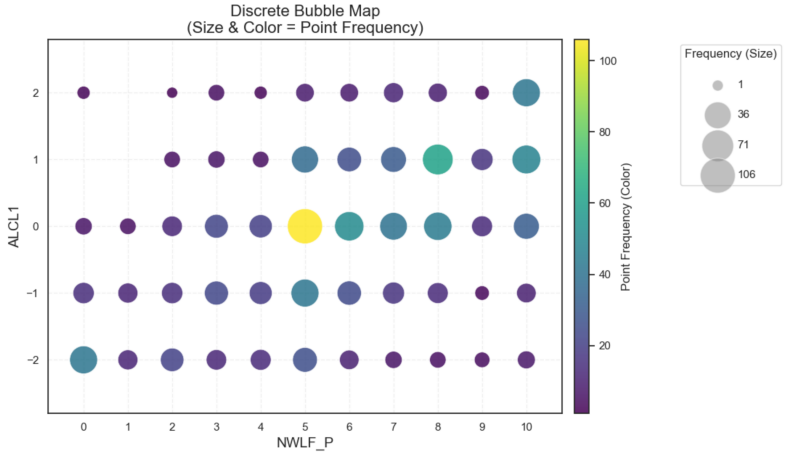


Figure 2: Bubble plot of the raw survey data responses for questions NWLF_P and ALCL1.

corresponds to the actual outcome of the voting questions

$$f = \{(x_i, y_j \,|\, i \in \mathcal{I}, j \in \mathcal{J})\}\,. \tag{2}$$

For a bubble plot of f, see Fig. 2. To turn the discrete information in the above into a continuous function, we renormalise f and apply a smoothing in order to obtain a density over the space, taking into account that the size of the sets $\mathcal{I}, \mathcal{J}$ may be different (as in Fig. 2).

1.2 Kernel density estimation

We use kernel density estimation (KDE) in *Python* which creates the following continuous function

$$P(X,Y) := \frac{1}{|\mathcal{I}||\mathcal{J}|h_x h_y} \sum_{i=1}^{|\mathcal{I}|} \sum_{j=1}^{|\mathcal{J}|} K\left(\frac{X - x_i}{h_x}\right) K\left(\frac{Y - y_j}{h_y}\right) \tag{3}$$

and allows us to pass from describing the specific discrete outcomes, to the joint distribution or the abstract probability function of the random variables X and Y. The standard Gaussian

kernel (which guarantees that the integral over the space gives probability 1) is

$$K(u) := \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}u^2\right). \tag{4}$$

In this way we arrive at a continuous approximation for the probability distribution of the marginals of two variables In the notation of Eq. (1)

$$P(X_1 = x_1, X_2 = x_2) \approx p(x_1, x_2) = \sum_{x_3,\cdots,x_M} p(x_1, x_2, \cdots, x_M) \tag{5}$$

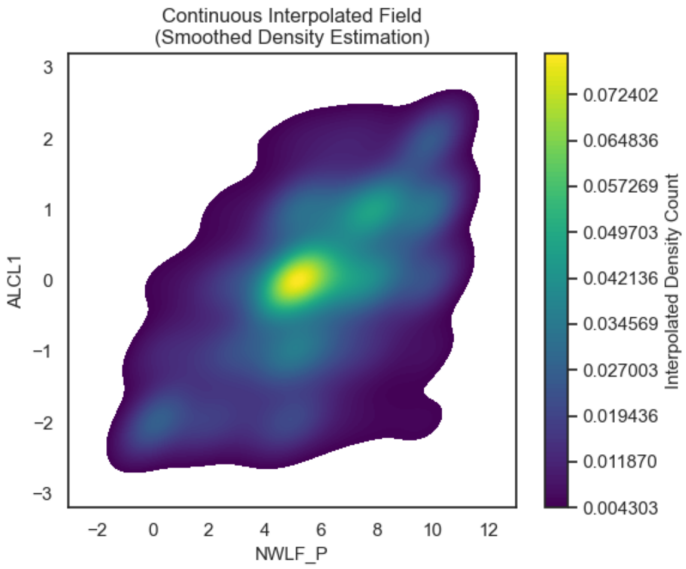


Figure 3: The transformation of f under a KDE, such as the one in Eq. (3). Notice that the function has been extended here beyond the original limits of the question responses.

For every pair of questions in the survey, there is a corresponding continuous probability distribution, which captures the distribution of responses; for M questions, there are $\frac{1}{2}M(M-1) \sim \mathcal{O}(M^2)$ pairwise probability density plots. For ~ 60 questions, this is roughly 3600 pairwise relation, i.e., 3600 distributions of the type depicted in Fig. 3; these are the distributions we train the machine to recognise.

2 Z as a predictor of the probability distribution between X, Y

At the end of the day, we are interested in finding out if the answers to some of the survey questions are good predictors of other responses. For instance, if we know somebody's age, can we predict how they will answer the rest of the survey? This has the effect of stratifying the population according to the conditioning; the continuous probability function in Eq. (3) becomes

$$P(X, Y|Z = z) := \frac{1}{|\mathcal{I}||\mathcal{J}|h_x h_y} \sum_{i=1}^{|\mathcal{I}|} \sum_{j=1}^{|\mathcal{J}|} K\left(\frac{X - x_i|z}{h_x}\right) K\left(\frac{Y - y_j|z}{h_y}\right), \tag{6}$$

where $x_i|z$, respectively $y_j|z$ mean $f_z = \Big\{(x_i, y_j \,|\, i \in \mathcal{I}, j \in \mathcal{J}, z)\Big\}$.

When conditioning, the corresponding bubble and smooth probability plot are updated, as shown in Fig. 4.

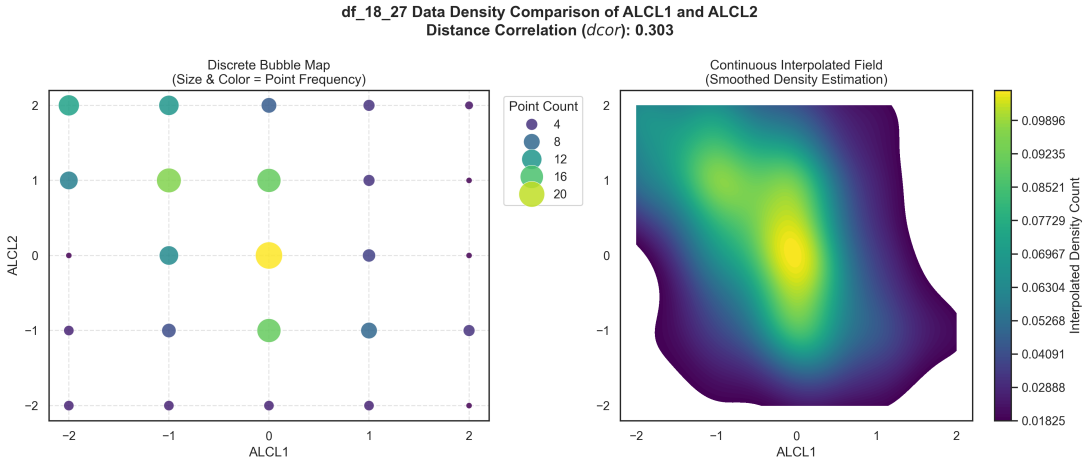


Figure 4: The raw data bubble plot and smoothing function for the questions $[ALCL1]$ and $[ALCL2]$ for the age group 18-27.

3 Synthetic data generating machine

We will first describe the desired device, before explaining how we will train the model. For clarity, we adopt the following notation to distinguish between data from the survey (denoted by voters $\mathbf{v}$) and synthetic voters produced by the machine (denoted $\mathbf{s}$)

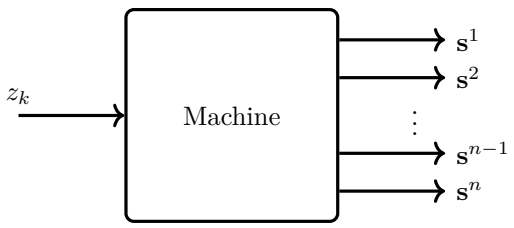
$$\mathbf{v}^j \in \mathbb{R}^M \qquad \text{full vector of survey responses for survey-data voter } j \tag{7}$$

$$\mathbf{s}^j \in \mathbb{R}^M \qquad \text{full vector of survey responses for synthetic voter } j. \tag{8}$$

We imagine the following scenario: a synthetic data generating machine receives an input x_i corresponding to a particular answer for question I (could also be a course-grained answer) to the questionnaire, such as "age range", "education level", or whatever else. On output, the machine produces n fake voters; each voter $\mathbf{s}^j$ is a vector of responses to the survey questionnaire with the question corresponding go variable Z fixed to be $Z = z_k$. We write the vector of survey responses for synthetic voter j with question K fixed to z_k as

$$\mathbf{s}^j(k) = (s^j_0, \cdots, z_k, \cdots, s^j_{M-1}) \in \mathbb{R}^M \tag{9}$$

and draw the machine schematically as an input-output system



The machine is re-run over all the possible responses to question I, i.e., the inputs is varied over all possible answers (or course-grainings) for that question $i \in \mathcal{I}$ in order to build-up a complete electorate with $(n \times |\mathcal{I}|)$ members

$$\{z_k\}_k \longrightarrow \mathcal{E}' = \{\mathbf{s}^j(k)\}_{k,j}. \tag{10}$$

3.1 Training

The machine M should, ideally, output an electorate $\mathcal{E}'$ that is close (in some statistical sense) to the electorate $\mathcal{E}$ in the original survey data. To do this, we train the machine to output each subpopulation $\{\mathbf{s}^j(k)\}_j$ for all $k \in \mathcal{K}$ such that each one is close in distance in some probability density plot pairs. That is, the output of the machine minimises the average distance between all possible probability density pairs

$$\min_{\{\mathbf{s}^j(k)\}_j} \frac{1}{\frac{1}{2}(M-1)(M-2)} \sum_{X,Y} D\Big(P^{\text{model}}(X,Y|Z=z_k), P^{\text{data}}(X,Y|Z=z_k)\Big) \qquad \forall\, k \in \mathcal{K}, \tag{11}$$

where the variables X, Y in P^{model} mean the random variables derived from applying the kernel method in (6) to the synthetic data $\{\mathbf{s}^j(k)\}$ output by the machine.

Example 1 (Eq. (11) in words for a concrete but made-up scenario.)**.** *For each age group in the set* $\mathcal{K} = \{(18-25), (26-34), (35-44), \cdots\}$*, output a group of* 50 *individual synthetic voters* $\mathbf{s}^j$ *for* $j \in \{1, \cdots, 50\}$ *that minimise the average distance* D *between all possible probability density question pairs. For the age group* $18-25$*, an instance of such a distance is*

$$D\Big(P^{Model}\Big([NSEC_P], [ALCL9]\Big|\, Age\ 18\text{-}25\Big), P^{Data}\Big([NSEC_P], [ALCL9]\Big|\, Age\ 18\text{-}25\Big)\Big),$$

and we wish to minimise the average over all possible feature pairs.

3.2 Distance measures for the loss

Let $p = P^{\text{data}}, q = P^{\text{model}}$.

Wasserstein-1 distance

$$W_1(p,q) := \inf_{\gamma \in \Pi(p,q)} \iint_{\mathbb{R}^2 \times \mathbb{R}^2} \|\mathbf{x} - \mathbf{x}'\|_2 \, d\gamma(\mathbf{x}, \mathbf{x}') \tag{12}$$

The algorithm searches over Π, which is the set of all joint distributions $\gamma(\mathbf{x}, \mathbf{x}')$ whose marginals are $p(\mathbf{x}) = p(x, y)$ and $q(\mathbf{x}') = q(x, y)$.

Kullback-Lieber divergence

$$D_{KL}(p \parallel q) := \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p(x,y) \log\left(\frac{p(x,y)}{q(x,y)}\right) dx\, dy \tag{13}$$

4 Conclusion

For the different synthetic data sets that one can create we are going to condition on any desired features. In our test we conditioned on:

$$Z = \{\text{AGE, (AGE and GENDER), ALCL10, ALCL6, ALCL9}\}. \tag{14}$$

and found that the best model always predicted best from (AGE and GENDER). The result of this deliverable is an input-output machine that allows the use to create a population of synthetic voters by specifying an age band for them as well as their gender. This allows one to build-up a synthetic population of voters that is demographically in-tune with the expected turn out for the city of Aarhus.

5 References

[1] Denis Boigelot. Correlation examples2. Wikimedia Commons, 2011. Licensed under Creative Commons CC0 1.0 Universal Public Domain Dedication. URL: `https://commons.wikimedia.org/wiki/File:Correlation_examples2.svg`.