Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

Vanja Falck

Synthetic Population Generation using Deep Generative Methods

Boosting Policy Games on Public Health

Master's thesis in Applied Computer Science
Supervisor: Chistopher Frantz
June 2023

NTNU
Norwegian University of
Science and Technology

Vanja Falck

Synthetic Population Generation using Deep Generative Methods

Boosting Policy Games on Public Health

Master’s thesis in Applied Computer Science
Supervisor: Chistopher Frantz
June 2023

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU
Norwegian University of
Science and Technology

Abstract

Deep generative methods have proven successful in producing synthetic populations with a significant number of attributes on individual data records. Synthetic populations can enrich, i.e., simulations in public health policy games with data close to real life. In addition, feature-rich synthetic populations support public governance by increasing data granularity to boost analysis on subgroups and in smaller locations. Finally, synthetic populations can mimic results from interventions to evaluate policies in games and real life. Given sufficient epidemiological strength, synthetic populations can support exploring real-life public health policy.

The main contribution of this project is to adapt state-of-the-art population generation by deep generative methods to the area of public health. If individual original data are available, deep generative methods provide robust and granular high-featured populations to, i.e. explore inequalities in health. However, the lack of explainability of outputs from neural networks adds to the existing lack of standards; measuring how well, i.e. the statistical patterns in original individual data reproduce in a high-featured synthetic population, is complicated.

This project proposes a quasi-experimental framework for assessing the quality of the synthetic populations, as the quality of synthetic populations has to be assessed in their particular and intended use context. This project aims to provide synthetic populations applicable to policy games teaching university-level students public health-related policy analysis and planning, particularly emphasising inequalities in health. Hence, this project evaluates the synthetic populations by examining differences and similarities from the original data and analyses differences in performance on a quasi-experiment using heterogeneous treatment effects from observational data. Comparing prediction outputs from the causal forest model is proposed as a viable external validation method for synthetic populations intended for use in analysis with health outcomes.

Contents

List of Figures

List of Tables

1 Introduction

Synthetic populations are datasets mimicking natural persons' data records that preserve an actual population's aggregated statistics. Synthetic populations have a wide range of use, including feeding attributes to agents in simulations. The first techniques for generating synthetic populations dated back to just before World War II and were used in analysing telecommunication. After the war, it gained broader usage as it, i.e. was a less expensive tool than collecting real-life data for governmental analysis [41, 21]. Current techniques for generating populations with few attributes are found within small area estimation and microsimulations [41] and agent-based modelling [22], which are re-weighting and synthetic reconstruction. While re-weighting requires individual original data and aggregated statistics, the synthetic reconstructions can create artificial data records from aggregated statistics only [41]. Microsimulation uses synthetic data records to create cross-tables for answering what-if questions often relevant to policy making [41], like how an intervention in a region will affect inequalities in health.

The techniques for synthetic population generation, like re-weighting and synthetic reconstruction, currently dominate the workflow for microsimulations, that is, first generate an initial population, then fit the population to aggregated statistics for a geographic area of interest, and third, allocate individuals into networks of interest, like households [6]. Unfortunately, these techniques only create high-quality synthetic data with a few attributes. More attributes create empty cells in the contingency tables that fit records to aggregated statistics. The empty cell problem is the ”curse of dimensionality”, as stated by Richard Ernest Bellmann in 1957 [41]. A recent development within transport research using agent-based simulation is to increase the number of attributes using dimensional-reducing deep generative methods to accomplish the first step in the microsimulation workflow [17]. The ”curse of dimensionality” deals with these data dimension reductions.

Variational autoencoders described by Kingma and Welling in 2014 [23] and generative adversarial networks suggested by Goodfellow [18], and made operative by Arjovsky et al. in 2017 [1], outperform current techniques when reproducing populations with a significant number of attributes [6] [17]. The new methods are called deep generative because their core is computational neural networks. Deep learning is currently a bouquet of black-box methods, often lacking clear interpretability and explainability [8], but proven excellent to create replicas of data like images, audio, texts and tabular data. The replication area is less sensitive to the consequences of black-box threats than models of predictions. However, replications embody some of the black-box elements that threaten the trustworthiness of predictions [8, 29]. For example, the replicas are mainly evaluated by humans or benchmarked against area-specific labelled datasets, which provide low transfer value between areas. The threat of, i.e. misrepresenting groups beyond any randomness could impact the analytical value of a synthetic population, as a tool for policy planning, by deep generative methods proposed in this project.

High-attribute synthetic populations created by deep generative methods are, to the writer's knowledge, yet to be used within the domain of microsimulation in public health. Such populations have yet to be evaluated regarding usefulness in policy analysis and planning within public health or education to improve skills in these fields. A recent review on microsimulation in public health [36] identifies 24 studies where the populations used were generated by either re-weighting or synthetic reconstruction. No reported studies in this review used deep generative methods. A general criterion for evaluating microsimulations with health outcomes is benchmarking them against known outcomes (if they exist), a principle that can be transferred to evaluating synthetic populations for policy analysis and planning in public health. Figure 1 illustrates the cross-over in relevant domains of this project.

The initial motivation for exploring high-attribute synthetic populations within public health is linked to the writer's work as an associate professor in social science and public health and the challenges of teaching university-level students to deal with the complexities of policy analysis and planning in combating inequalities in health. One way of enhancing skills in the area is to play with interventions and assess their impact on a particular population over time. Therefore, the context for generating synthetic populations in this project is set to a simple policy game [26] in the making for analysis and planning within public health with particular emphasis on reducing inequalities in health [20]. The choice of context serves two purposes for evaluating high-attributed synthetic populations. First, the synthetic populations can support authentic gameplay, which provides learning without heavy support from teachers and access to populations replacing sensitive individual data, for example, for health outcome analysis. Second, synthetic populations assisting gameplay close enough to stage real-life scenarios with epidemiological rigour are, by definition, regarded as authentic. High attributed synthetic populations embedded in a game world with policy challenges of inequality in health additionally allow for a spillover of experiences relevant to epidemiological simulations without promising a thorough evaluation covering rigorous epidemiological demands. As a simpler proxy for the epidemiological evaluation of a synthetic population, the heterogeneous treatment effects calculated by causal forest [2, 16] is proposed.

This project reproduces the deep generative methods of creating synthetic populations from transport research [6] [17] on EU-SILC living condition data from Finland and Norway. The EU-SILC data includes reported self-perceived health, used as the health outcome in evaluating the populations. Furthermore, as neural networks are known to be notoriously insensitive to underlying statistical properties [24], a self-supervised contrastive clustering algorithm based on the maximum coding rate reduction [49] is applied to the original and synthetic data for comparison of cluster patterns.

In addition to this project's primary task of generating and evaluating synthetic populations for ed-

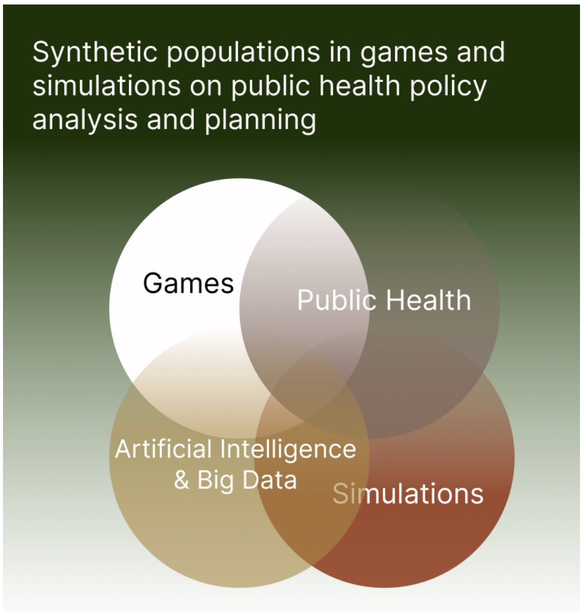

Synthetic populations in games and simulations on public health policy analysis and planning
Games
Public Health
Artificial Intelligence & Big Data
Simulations

Figure 1: Areas cross-overs in the contextual embedding of synthetic populations in a policy game on public health

ucational public health purposes, experiences with the deep generative models and heterogeneous treatment effects are shortly discussed in light of their applicability as potential components in a policy game.

1.1 Research Questions

This project responds to the identified gaps and the opportunities for applications in the area of public health by answering the following research questions:

Q1: How can state-of-the-art deep generative methods contribute to creating high-attribute synthetic populations from original individual data for use in policy games on public health?

Q2: To what extent can deep generative techniques be exploited to artificially increase the granularity of high-attributed synthetic population data to investigate small groups and locations to support analysis and planning in public health, emphasising reducing health inequalities for educational purposes in a policy game?

Q3: How can the high-attributed synthetic populations be evaluated according to their applicability in a policy game on public health?

Q3a: How can the high-attributed synthetic populations be evaluated to meet the educational requirements in a policy game on public health?

Q3b: How can the high-attributed synthetic populations be evaluated to meet the epidemiological requirements in an exploratory real-life-oriented policy game on public health?

1.2 Keywords

- synthetic populations
- deep generative methods
- policy game
- public health
- epidemiological simulation
- causal forest

1.3 Audience

The audience for this work is diverse. Game developers interested in public health-related themes or using synthetic populations to embody agents in settings that require some form of outcome evaluation will benefit from the code and experiences shared in the project. Educators within the public health and epidemiology field can seek inspiration from ideas related to using synthetic populations in general outcome analysis and policy planning or to inspire to initiate new serious games using synthetic populations. Epidemiologists can be inspired to sharpen the evaluation and develop methods for even better synthetic populations fitted for real-life health outcome analysis. People working with governmental planning in public health or public data access can use the tentative evaluations relevant to health outcomes analysis based on synthetic populations created by deep generative methods.

1.4 Structure

The following text starts with an overview of related work with particular emphasis on the deep generative methods and validity, as the main focus in this work is the generation and evaluation of synthetic populations. Next follows a brief overview of the areas in which this work is embedded. That is, policy games and public health, with particular emphasis on social determinants for health that are central to combat inequalities in health. The next chapter covers the methods, including the initial literature review on recent contributions to synthetic population generation. This literature review is extended based on ’snowballing’ (i.e., augmentation with additional works found in documents identified as part of the systematic search) to gain a broader perspective and return to contributions before the selected time frame. Besides a general description of methods, this chapter also briefly introduces how synthetic populations for use in public health policy analysis and planning can be validated in the context of quasi-experiments. This is followed by chapters describing and analysing the results of the experiments, an associated discussion and a conclusion. Finally, as the audience of this work is diverse, and some readers need insight into the operational details, comprehensive documentation of code and visuals for all single variables in the different versions of synthetic populations are provided in the appendix.

2 Related Work

The proposed deep generative methods for population synthesis [6] and generative adversarial networks [17] belong to self-supervised learning within machine learning. A known weakness of self-supervised neural networks is that they, by default, do not necessarily respect the statistical patterns of the original data [24]. Self-supervised contrastive learning to identify clusters and, therefore, hidden statistical patterns in data are suggested to add an explanation to the black-box outcomes from neural networks [24] [49] like the synthetic populations in this project. Self-supervised clustering is run on original and synthetic populations. However, the implications of different or almost similar patterns still need clarification.

The synthetic populations created in this project are contextually framed in public health and, in particular, embedded in an educational policy game in the making, focusing on policy analysis and planning to reduce inequalities in health. Public health and epidemiology are closely related but only partially overlap. The evaluation of synthetic populations by quasi-experiment on heterogeneous treatment effects belongs partly to epidemiology and partly to tree-based machine learning.

The following related work is briefly described, and a short introduction to policy games and public health is provided. Lastly, the synthetic populations need to be evaluated in some specific context related to their proposed use area. Therefore, a setup of a quasi-experimental design to evaluate synthetic populations is suggested.

2.1 Deep Generative Methods for Population Synthesis

2.1.1 History of Synthetic Population Generation

A synthetic population is a collection of members with statistical properties similar to members from the natural population, "...and possibly such that, at a later stage, the synthetic population can be aligned with attributes that represent future targets." The latter is framed by Muller and Ivt (2011) in the context of agent-based simulations. In this project, this definition is adopted with the following twist "...and possibly such that, during various experimental or quasi-experimental analyses on health outcomes, the synthetic population reveal similar patterns of results."

Before entering the arena of population generation in the area of microsimulations, the reader should note that population generation, as such, is not the central focus of microsimulations. Instead, the initiation and subsequent change in member attributes are in focus when some external event occurs, like introducing a new public health intervention such as offering youths from low-income families free equipment and access to public leisure and sports activities. The populations should fit the "what would happen if..." scenarios because such questions are relevant to policy analysis. The populations should be ready for cross-tabulation to generate data that could occur if the simulation was a real-life event. This approach differs from this project's aim at evaluating the population as such but also overlaps, as this project also engages in the generated population's ability to respond similarly to original data in the given quasi-experimental settings.

Two frequently used methods for population synthesis for microsimulation and public health, in particular, are variants of iterative proportional fitting and combinatorial optimisation [36]. Both methods belong to the synthetic reconstruction branch, while iterative proportional fitting also can fit in with the re-weighting branch. Moreover, they can use individual survey data and extend a synthetic population to a geographic region with information on its aggregated demographic statistics [36]. Various microsimulations with health outcomes like diet, body weight, smoking, gambling, mental and dental health, diabetes and mortality (ref Smith) use these two methods.

This project is not concerned with merging survey data with aggregated statistical patterns for a particular geographical region nor creating a population based on aggregated statistics only. Instead, the interest is in the increasing number of available individual socio-economic and health-related observational data and how these data can translate into synthetic populations for use, i.e. policy analysis and planning and, in this project's case, an educational policy game on public health. Nevertheless, much knowledge gained in microsimulation regarding evaluating synthetic populations is valuable for synthetic populations generated by deep generative methods pipelined to quasi-experiments to obtain health outputs. However, this project aligns with the deep generative methods used for making synthetic populations for agent-based modelling in transport research based on complete original individual data.

2.1.2 Variational Autoencoder and Generative Adversarial Networks

This study uses a generative adversarial network first suggested by Ian Goodfellow in 2013 [18] and later improved by Arjovsky in 2017 [1], and the variational autoencoder presented by Kingma and Welling in 2014 [23]. These methods are self-supervised because they learn from attempts to create a replica of its input. The generative adversarial network learns from a competition between two neural networks, a generator creating replicas and a critic evaluating those replicas and deciding if they are fake. The variational autoencoder learns by calculating probabilities on a latent representation of data. First, data is pushed through a neural network encoder to produce the latent representation. Then an operation is done on the latent representation to calculate probabilities before the output from the encoder transfers through a new neural network, often a mirrored version of the encoder, called the decoder. The output from the decoder is the replica. The input to both models is random samples in the dimension of the latent representation taken from a Gaussian distribution. These networks are explored within transport research, and the models from these studies [6, 17] are only slightly modified for use with strictly one-hot-encoded categorised variables on living condition data from EU-SILC for Finland and Norway in this project. An essential difference between the transport research articles and this project is that the synthetic populations must fit into different quasi-experiments. The public health experiments look for health outcomes over time, while the transport research looks for travel forms and paths in a spatiotemporal space. In a public health policy setting, the synthetic population must therefore lend itself to matching experiments with individual health outcomes and not a sum of collective agents' actions of choosing a particular geographical path from place A to place B at a particular time, as in the transport research. Hence, the evaluation of the synthetic populations differs. This project makes particular efforts to check if the synthetic populations can replicate actual data results from the quasi-experiment of heterogeneous treatment effects.

2.2 Policy Games

A policy game [26] is a game or simulation that explores a complex field like public health [38] with many actors interacting across many areas of society, with its core player activities being policy analysis and planning with or without collaborative challenges. In this project, the synthetic populations are contextually embedded in the frame of a policy game on public health, where these populations are the basis for the analysis and planning of interventions to reduce inequality in health. The analysis part is wrapped in a quasi-experimental frame in which the synthetic populations are input. The policy planning part is related to the quasi-experiment outputs, which pinpoint from analysis which subgroups would benefit from which interventions.

Policy games can be serious games [44] in the sense of serving a purpose outside the gameplay itself, like an educational goal. Igor Mayer defines simulation games as "expert (m)ent(i)al, rule-based, interactive environments, where players learn by taking actions and by experiencing their effects through feedback mechanisms that are deliberately built into and around the game. Gaming is based on the assumption that the individual and societal learning that emerges in the game can be transferred to the world outside the game" [26].

A review paper looking for studies on games using simulations published between 2001 and 2020 found 52 articles demonstrating 14 different areas, including public health [39]. Unfortunately, the two articles with games on public health were not referenced, and neither were retrieved by the attempt to search. Urban planning was the most frequent field. Agent-based modelling has been used to simulate the development of inequalities [4] and access to resources relevant to public health, like access to food, [45], and food security [42]. These contributions show how interventions, through simulations, technically can be played out in a policy game by using synthetic population data to embody agents as non-player characters to simulate scenarios related to the chosen intervention. Simulations have been used extensively in epidemiology and to inform health policies, as shown in a recent review by Jalali et al. [21].

Several policy games aim to get stakeholders together to fight or negotiate for their interests to solve a higher common goal, often a wicked problem like reducing inequalities in health [38]. In this project, the focus is not on collaboration but on how high-attribute synthetic populations can contribute to policy exploration and, more specifically, how synthetic populations can be evaluated to meet requirements held by such explorations. Figure 2 suggests a simple policy game setup. The synthetic population is the point of departure, as it provides the player, as a policy maker or stakeholder, with information on the inhabitants of a region. The quest is to develop or identify interventions that can be implemented in this region to reduce the current inequality in health. The interventions can be simulated, either passively, that is, by providing a new set of population data given a particular intervention, or dynamically by running modelled simulations on the population data to explore outcomes on health for different groups of inhabitants. The gameplay is used as a context for diving into producing and evaluating high-attributed synthetic populations for public health purposes and is not a part of this project.

A policy game can explore the consequences of changes in such structural features for different groups of citizens. WHO defines public health as ”...an organised effort by society, primarily through its public institutions, to improve, promote, protect and restore the population’s health through collective action. It includes services such as health situation analysis, health surveillance, health promotion, prevention, infectious disease control, environmental protection and sanitation, disaster and health emergency preparedness and response, and occupational health, among others.” [28]. Policy related to public health (health in all policies) is defined ”...a policy or reform designed to secure healthier communities, by integrating public health actions with primary care and by pursuing healthy public policies across sectors.” [28]. Health policy is defined as ”... set of decisions or commitments to pursue courses of action aimed at achieving defined goals for improving health, stating or inferring the values that underpin these decisions...” [28] . To summarise and reformulate, the definitions of public health policy used in this project are the laws, regulations, plans, decisions and actions implemented within society to promote wellness and ensure that specific health goals

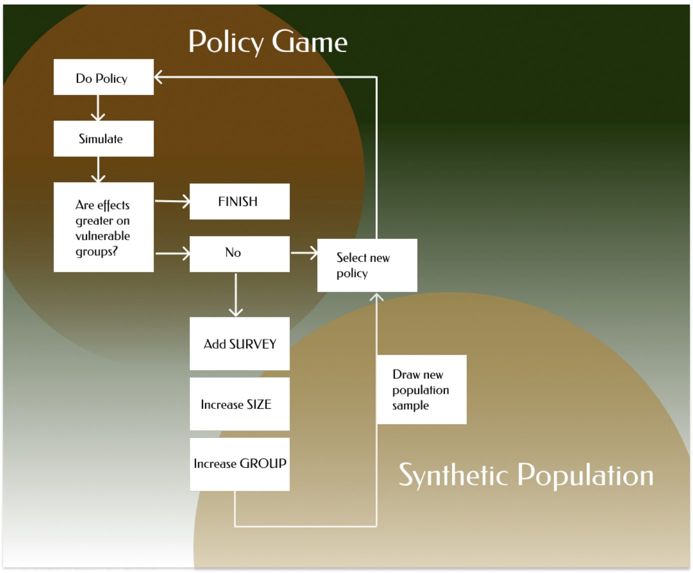


Figure 2: Simplified simulation using synthetic population techniques in a policy game to reduce inequalities in health

are met.

Simulations by, for example, simply manipulating individuals' data records [37] or playing out complex agent-based strategies [35] are significant areas for the use of low-featured or non-demographic synthetic populations within public health. In this project, synthetic populations are *high-featured* and so-called *demographic*. Demographic means populations should mimic population distributions found in real life to be relevant to public health policy analysis and planning. Synthetic populations with sufficient epidemiological quality can transform an educational policy game into a tool for governmental officials and planners to design realistic scenarios.

2.3 Public Health

The World Health Organisation states: ”Public health is an organised effort by society, primarily through its public institutions, to improve, promote, protect and restore the population's health through collective action.” In the case of inequality in health, the concept of determinants of health has been well established [11, 10].

Thirty years ago, Margareth Whitehead and Gøran Dahlgren [11, 10] sketched out the ”Rainbow model” on determinants of health in figure 4. In a recent review [12], they elaborate on the future use of their model. In contrast to determinants of health, epidemiology looks for the absence of disease. The rainbow model highlights areas where health is created and encourages cross-sectional policy work. Public health and healthcare systems cannot deal with upholding health and, mainly,

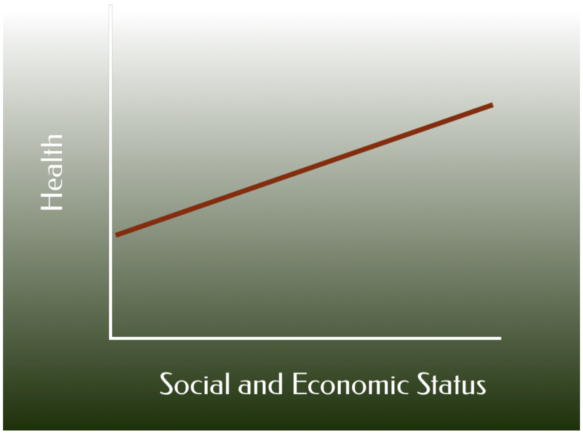


Figure 3: Social and Economic Gradient in Health

equality in health. Promoting equality in health requires broad actions across every governmental sector. The rainbow model sketches these areas, mainly outside the healthcare domain. According to Whitehead and Dahlgren [12], their model describes well but fails to deal with inequalities in health. To mitigate this, they suggest complementing social determinants of health with, i.e. the Diderichsen Framework, which allows for explaining pathways and mechanisms [13] [33]. Diderichsen's framework has four main tools operating on the determinants of health. The first is differential power and resources. The second is differential exposure, the third is differential vulnerability, and the fourth is differential consequences of being sick [12].

In this project, the generated population should idealistically allow simulations with any health status as outcomes [5, 25]. Determinants for health and the mechanisms and pathways changing those determinants are the primary independent variables in the individual data records of the population. Municipalities dealing with inequality in health (fig. 3) addressed by "Folkehelseloven" [20] is the thematic focus for the policy game. In this project, health outcome is self-perceived health, a measure available in EU-SILC for Finland and Norway [15]. This health outcome is framed with other variables describing individuals' social and economic status in EU-SILC data on living conditions, data that finally provides person records data to the synthetic populations.

2.4 Quasi-Experiments

The term quasi-experiment was first termed by Stouffer and later by Campbell [9, p. 6] to define experiments that have treatments, outcome measures and experimental units but lack the random assignment of randomised controlled trials just like the heterogeneous treatment effect based on observational data, like EU-SILC that is used in this work.

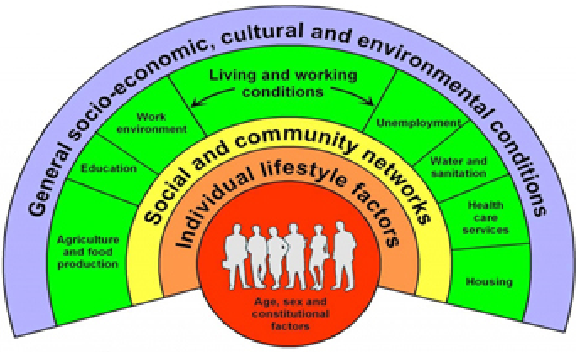

General socio-economic, cultural and environmental conditions
Living and working conditions
Work environment
Education
Agriculture and food production
Unemployment
Water and sanitation
Health care services
Housing
Social and community networks
Individual lifestyle factors
Age, sex and constitutional factors

Figure 4: Whitehead and Dahlgren's Social Determinants for Health

The evaluation of synthetic populations for use in policy analysis and planning in public health should comply with specific equivalence standards if they replace original data in the analysis. In this project, a data-driven pipeline for a quasi-experiment [9], implying a study not complying with the strict standards of a randomised controlled trial which usually is the case for field studies, is set up to evaluate the synthetic populations.

The quasi-experiment uses the observational EU-SILC data and the variable self-perceived health as the outcome variable. Observational data can be used for calculating heterogeneous treatment effects [2] by selecting some variables as intervention variables. Such methods are used as pilot studies before an actual trial or as a self-contained tool for analysing trends in larger groups. The setup is advantageous in identifying diverse outcomes among subgroups following an intervention [2]. Such design, therefore, particularly fits this project which seeks opportunities for synthetic population analysis relevant to inequalities in health. First, the method is a good candidate for evaluating the analytical equivalence when health outputs from actual or imagined experiments are at stake. Second, the heterogeneous treatment analysis is an excellent tool for identifying interventions that fall differently between vulnerable and non-vulnerable groups. A side effect is that the heterogeneous treatment effect method can be used directly as a mechanic in a policy game to decide an agent's future response to an intervention, given its attributes from the synthetic population.

2.5 What and Why

This project creates high-attribute synthetic populations from data on living conditions from the EU-SILC data for Finland and Norway. Methods to compare the general statistical patterns in replica with the original are gathered from the microsimulation and small-area-estimation, making a low-attribute synthetic population. Self-supervised clustering describes possible differences between the original and synthetic populations. However, this work is a preliminary suggestion only. Lastly, the quality of the synthetic populations is suggested and assessed by the replica's ability to reproduce similar outcomes from heterogeneous treatment analysis.

Educational Relevance Theoretically, the policy game's educational stance on experimenting and evaluating policies is rooted in experience, problem and case-based learning [32] and scaffolding [40] [46]. However, while epidemiological relevance requires the synthetic populations to be close to reality, a policy game's educational value [27] can sometimes do without the same level of alignment with reality to gain sufficient academic authenticity. In education, an utterly fake toy problem can probe learning. In this project, the envisioned illustrative policy game aims to increase students' understanding of complex social and structural phenomena and skills for developing interventions to improve health and reduce health inequalities in a population. While developing such skills, statistical literacy is necessary. The latter can be accomplished with any synthetic population without substantial similarities with the original data, as data can be taken for base truth without losing anything in the following step of learning the computational skills of analysis. As sketched out, a policy game should also support an understanding of interventions to reduce inequalities in health. Therefore, synthetic populations must preserve some critical epidemiological properties to emulate the realistic effects of explored interventions.

3 Material and Methods

Having established the background and motivational use case of the generation of synthetic populations, the following discusses the origin and nature of the data on which the exploration of techniques performed as part of this research is based.

3.1 Original Data

Data on welfare and social attributes are freely available as synthetic individual data records from most EU countries like Finland [15] that is used in this project. The Norwegian EU-SILC data was downloaded from SIKT [34] after being granted access for use in this project. The Finnish EU-SILC, a synthetic dataset, and the Norwegian EU-SILC, an original dataset, are treated as original data in this project. Therefore, no interpretations of selection and weighting of people or questions related to, i.e. operationalisation of variables are discussed. Instead, the data are taken for their face values when input to the workflow of generating synthetic populations and later when analysing quasi-experiment outcomes.

The Finnish data are for 2013 with 19291 examples, and the Norwegian data are from 2017-2020 and merged into one dataset with 24720 examples. Both datasets imputed missing values on variables

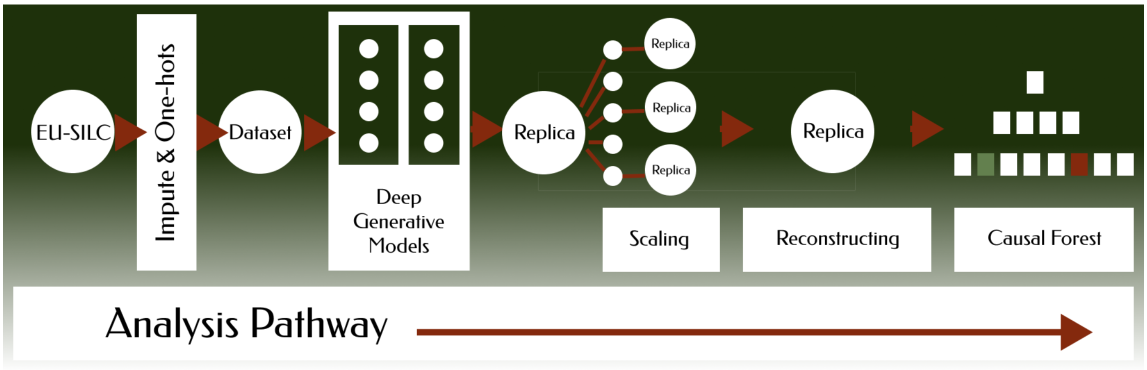


Figure 5: Analysis pathway starting with data cleaning and transforming variables to one-hot-encoded or binary features. The prepared dataset is made into synthetic populations by two deep generative methods, variational autoencoder and Wasserstein generative adversarial network. The original and synthetic data are clustering using a variational autoencoder as an augmentor for contrastive self-supervised clustering using the neural manifold clustering and embedding setup. Clustering patterns on features and particularly the outcome variable ”self-perceived health” are used to enhance the similarity in results from Causal Forest, which measures the heterogeneous treatment effect from observational data.

if the number of missing was less than half for the Finnish data and less than 40 per cent for the Norwegian data. Suppose these data were to be used in an actual analysis rather than as here only as a substitute. In that case, this level of imputing missing variables could be challenging to some interpretations of results. However, these considerations are ignored, as running any epidemiologically interpretable analysis is irrelevant. In this project, the imputed EU-SILC data is taken for truth; it is assumed they are already prepared to fit epidemiological analysis. In this way, the EU-SILC data from Finland and Norway are considered the accurate epidemiologically relevant baseline. The next step is to compare these originals with the synthetic populations' abilities to, i.e. produce similar results from the same quasi-experiments.

Notice also that the weighting information for each example is not used as an asset in the generation of populations for the same reasons as sketched above. It does not matter if the data are ”real” and prepared for epidemiological analysis, as the assumption is that the data at hand is prepared. The point is not to predict, i.e. health outcomes, but to measure the reproduction quality of the synthetic population from some base ”truth”.

Self-reported health, variable PH010 with five categories from 1 to 5, where 1 is excellent health, is used as the outcome variable in quasi-experiment analyses for both datasets. Self-reported health is tightly connected to physical and psychological health and illnesses. Therefore, it is used within public health and epidemiology as a reasonably good indicator of health at the population level. The datasets do not contain the same variables, even though some overlap exists. This is because the Finnish dataset was quickly available for download and was used as input to the initial explorations of the deep generative methods. Only later, when the Norwegian dataset was available, making the two sets identical became challenging because the synthetic EU-SILC had a different setup and ways to present the variables and different variable names. No code-book translation to the synthetic EU-SILC was available through the Norwegian Statistical Bureau or SIKT. However, it is not particularly important to this project that the datasets match all variables. They represent an actual population when the variables' values are taken to represent the population, and both have the same output variable, self-perceived health, which is essential for the quasi-experiment evaluation.

3.1.1 Cross-Sectional Data - EU-SILC Finland 2013

The EU-SILC data [15] is cross-sectional on income, poverty, social exclusion and living conditions and can be used to generate and evaluate cross-sectional population generation. Data was obtained from Euro-Stats Website in October 2022 with coded variable names. All variables used in the population data from Finland are translated to binary or one-hot-encoded categorical variables (see variable list below 1). In addition, all variables, including numerical ones, are transformed to categorical one-hot-encoded, resulting in 230 binaries from 38 variables.

Variables with only two options in the data, like gender, are transformed into a one-column binary. For example, numerical float data has variables for income and benefits. These are summed up and turned into two binary variables, ”hasIncome” and ”hasBenefits”. One is assigned if the total sum is positive. The category year of birth is a numerical integer turned into a five-category ordinal variable ”Age”. The division of age starts from the youngest person in the data, 17 in 2013, the year of the observation, and picks groups spanning 13 years up to the last age group of 82 or older. All categorical variables are treated as nominal, and all non-binaries are, therefore, one-hot-encoded. Some categorical variables are ordinal and could have been made as scale variables with no problem running this in the generative models. Experiments on the advantages or drawbacks of using scaled versus one-hot-encoding on ordinal variables are not part of this work and are left to future research. The one-hot-encoding strategy will work on any variable. This strategy is selected because social variables needed to understand inequalities in health often are ordinal.

#	Description EU-SILC Finland	Original Type	Values	Name
1	Household size	numerical (int)	5	householdSize
2	Year of birth	numerical (int)	5	Age
3	Gender	binary	2	isFemale
4	Marital status	categorical	5	PB190
5	Education level	categorical	5	PE040
6	Economic status	categorical	11	PL031
7	Work	categorical	4	PL040
8	Self-perceived health	categorical (scale)	5	PH010
9	Long term health problem	binary	2	hasIllness
10	Activity reduction	categorical	3	PH030
11	Access to healthcare	categorical	3	PH040
12	Access to dental care	categorical	3	PH060
13	Buy new cloths	categorical	3	PD020
14	Two pair shoes	categorical	3	PD030
15	Get together friends	categorical	3	PD050
16	Leisure	categorical	3	PD060
17	Spend money personal	categorical	3	PD070
18	Internet at home	categorical	3	PD080
19	Life satisfaction	categorical (scale)	10	PW010
20	Meaning of life	categorical (scale)	10	PW020
21	Economic satisfaction	categorical (scale)	10	PW030
22	Accommodation satisfaction	categorical (scale)	10	PW040
23	Nervous	categorical (scale)	5	PW050
24	Feeling down	categorical (scale)	5	PW060
25	Calm	categorical (scale)	5	PW070
26	Depressed	categorical (scale)	5	PW080
27	Being happy	categorical (scale)	5	PW090
28	Satisfied with time use	categorical (scale)	10	PW120
29	Trust political system	categorical (scale)	10	PW130
30	Trust in legal system	categorical (scale)	10	PW140
31	Trust in police	categorical (scale)	10	PW150
32	Satisfied personal relationships	categorical (scale)	10	PW160
33	Someone to discuss personal issues	binary	2	hasFriend
34	Help from others	binary	2	getHelp
35	Trust in others	binary	2	PW190
36	Satisfied green areas	categorical (scale)	10	PW200
37	Satisfied living area	categorical (scale)	10	PW210
38	Physical security	categorical (scale)	4	PW220
39	Income	numerical (float)	2	hasIncome
40	Benefit	numerical (float)	2	hasBenefits

Table 1: Variables from EU-SILC Finland

3.1.2 Cross-Sectional Data - Norwegian Income and Living Conditions 2017-2020

The Norwegian official welfare data are similar to synthetic EU-SILC [15] for Finland, having some of the variables but is not entirely the same as for Finland 2013. The dataset has partly general EU codes and partly Norwegian variable names. In addition, some variables overlap with the Finnish data. Therefore, the two datasets cannot be directly compared in the following quasi-experiments.

All variables with two options are translated to a one-columns binary variable. All other categorical variables are one-hot-encoded. Numerical variables like age are turned into ordinal categorical variables and next to one-hot-encoded. The economic variables extracted from the original EU-SILC Norway are at the household level and transferred into a binary ”hasIncome” and ”hasBenefits”, giving one if the sum is positive and zero otherwise. In addition, the general EU-SILC poverty indicators are included, such i.e. affording health and dentistry, new clothes and shoes, leisure, spending time with friends, and using own money.

Socio-demographic variables are gender ”isFemale”, ”Age”, marital status (PB190), level of education (PE040), ”hasKids”, ”livesAlone”, ”householdSize”, region of residence, and number of inhabitants in place of residence.

In concert, the selected social and economic variables comprise some of the determinants of health that are important for investigating inequalities in health.

#	Description EU-SILC Norway	Original Type	Values	Name
1	Household size	numerical (int)	9	householdSize
2	Year of birth	numerical (int)	5	Age
3	Gender	binary	2	isFemale
4	Marital status	categorical	5	PB190
5	Education level	categorical	6	PE040
6	Economic status	categorical	11	work
7	Economy	categorical	6	economy
8	Self-perceived health	categorical (scale)	5	PH010
9	Long term health problem	binary	2	hasIllness
10	Activity reduction	categorical	3	PH030
11	Access to healthcare	categorical	3	PH040
12	Access to dental care	categorical	3	PH060
13	Buy new cloths	categorical	3	PD020
14	Two pairs of shoes	categorical	3	PD030
15	Get together friends	categorical	3	PD050
16	Leisure	categorical	3	PD060
17	Spend money personal	categorical	3	PD070
18	Internet at home	categorical	3	PD070
19	Life satisfaction	categorical (scale)	10	PW010
20	Meaning of life	categorical (scale)	10	PW020
21	Feeling happy	categorical (scale)	10	AffectA
22	Worried	categorical (scale)	10	AffectB
23	Feeling sad	categorical (scale)	8	AffectC
24	Afford housing	binary	2	canPayHousing
25	Afford unexpected expense	binary	2	hasCapasity
26	Over mean income	binary	2	overMedianIncome
27	Income	numerical (float)	2	hasIncome
28	Capital income	numerical (float)	2	hasCapitalIncome
29	Benefit	numerical (float)	2	hasBenefits
30	Lives alone	binary	2	livesAlone
31	Lives in a city	binary	2	livesCity
32	House type	categorical	4	houseType
33	Has a PC	categorical	3	PC
34	Has a Internet	categorical	3	Internet
35	Has a car	categorical	3	Car
36	Inhabitants in place of living	categorical	4	sizePlace
37	Region in country	categorical	6	region

Table 2: Variables from EU-SILC Norway

3.2 Literature Review on Synthetic Populations

A systematic critical literature review [47] to investigate the field of synthetic population generation within the (social) simulation area is done. Oria and Google Scholar were used on September 19 2022, using the terms ”synthetic population”, ”generation”, and ”methods”. In addition, peer-reviewed papers published from 2018-2022 were selected. The selection criteria are described in figure 6. Fifteen articles were selected for reading, resulting in ten being rejected. The five remaining and included studies are listed in table 3.

Chapuis et al. [22] review articles published in the Journal of Artificial Societies and Social Simulation on methods of creating synthetic populations in social simulations. They divide the traditional heuristic techniques reviewed into two main approaches, combinatorial optimisation and synthetic reconstruction.

Yameogo et al. use synthetic reconstruction on French survey and census data to test different traditional heuristic methods to create a two-level synthetic population on households and individuals [48]. Hierarchical iterative proportional fitting and relative entropy minimisations are the currently best methods to combine individual and household-level populations. This study uses all three stages of traditional synthetic population generation; a) prepare a starting population of individual data records from survey data, b) fit individual data to aggregated statistics using census data, and c) spatial allocation of individuals and households. Similarly, Roszka et al. [31] create a population using Polish census (EU-SILC data) and survey data to investigate inequalities in income spatially. Finally, this article discusses the general validation of generated populations of relevance for this project.

The following two studies using deep generative methods in population synthesis mainly explore the first stage of preparing individuals’ data records, leaving the last two more or less unexplored. Borysov et al. created a population with variational autoencoding [6]. Their model matched the performance of the traditional iterative proportional fitting on data with limited attributes. However, with significantly more features, their model did better. Farooq et al. used generative adversarial networks to create a population based on Danish travel data [3]. They compared their generative model to variational autoencoders [6] and showed further improvements. These articles are related to transport research. Nevertheless, these techniques apply to any generation of individual data records. Neither of the presented studies aims at reproducing longitudinal population data.

This study takes advantage of previous discussions on methods and metrics of internal validation of synthetic populations. While the transport research on deep generative techniques only focuses on the first of the three stages of population synthesis, this study will include challenges in the second and third stages focusing on up-scaling data to fit the general population profile of an actual

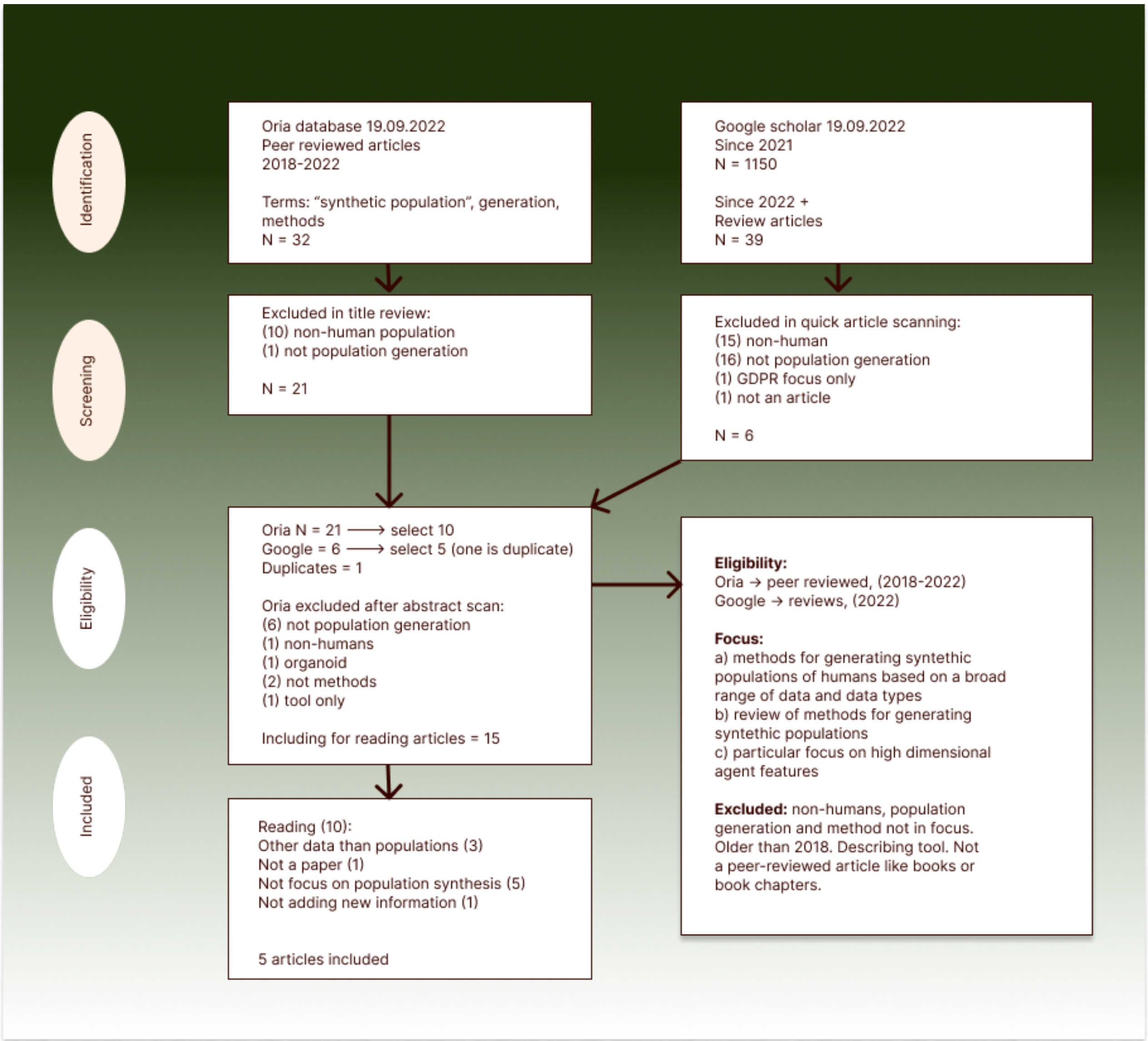

Identification
Screening
Eligibility
Included
Oria database 19.09.2022
Peer reviewed articles
2018-2022
Terms: "synthetic population", generation, methods
N = 32
Google scholar 19.09.2022
Since 2021
N = 1150
Since 2022 +
Review articles
N = 39
Excluded in title review:
(10) non-human population
(1) not population generation
N = 21
Excluded in quick article scanning:
(15) non-human
(16) not population generation
(1) GDPR focus only
(1) not an article
N = 6
Oria N = 21 ⟶ select 10
Google = 6 ⟶ select 5 (one is duplicate)
Duplicates = 1
Oria excluded after abstract scan:
(6) not population generation
(1) non-humans
(1) organoid
(2) not methods
(1) tool only
Including for reading articles = 15
Eligibility:
Oria → peer reviewed, (2018-2022)
Google → reviews, (2022)
Focus:
a) methods for generating syntethic populations of humans based on a broad range of data and data types
b) review of methods for generating syntethic populations
c) particular focus on high dimensional agent features
Excluded: non-humans, population generation and method not in focus. Older than 2018. Describing tool. Not a peer-reviewed article like books or book chapters.
Reading (10):
Other data than populations (3)
Not a paper (1)
Not focus on population synthesis (5)
Not adding new information (1)
5 articles included

Figure 6: Literature Search Article Selection

municipality.

The first literature search and selected articles led to a term similar to synthetic populations: small area estimation methods used to generate individual data records. A new search in Oria on February 1 2023, using keywords:” small area estimation”,” data”, ”generation”, and” methods” for the years 2018 until now that is peer-reviewed returned eight articles.

In addition to the systematic critical review, two unsystematic narratives [19] searches in Oria were done as part of so-called snowballing by, i.e. nesting in references in included articles. Titles and abstracts were scanned for relevance. The first search on September 22 2022, with the words **generative adversarial networks ”synthetic population”** method, did not give any new hits. The second search on policy games in public health, October 7, 2022, using the terms **”policy game” public health**, gave 26 unique peer-reviewed articles. About half were irrelevant by title. The most relevant articles are described in the section ”Related work”.

Incl.	Selected Articles by Title	Year/Author
Yes	How to generate micro-agents? A deep generative modelling approach to population synthesis	2019/Borysov [6]
Yes	Spatial microsimulation of personal income in Poland at the level of subregions	2019/Roszka [31]
Yes	Generating a two-layered synthetic population for French municipalities: Results and evaluation of four synthetic reconstruction methods	2021/Yameogo [48]
Yes	Generation of synthetic populations in social simulations: a review of methods and practices	2022/Chapuis [22]
Yes	Composite Travel Generative Adversarial Networks for Tabular and Sequential Population Synthesis	2022/Farooq [3]

Table 3: Selected Articles

3.3 Deep Generative Methods

The deep generative methods are built up by neural networks in an architecture for self-supervised learning. Neural networks are function approximators that exploit the backpropagation of errors through complex networks or neurons in layers to learn by updating weights and biases on each neuron. The errors are measured with various loss functions, which have to be differentiable to track small changes in error. The two methods used in this project operate differently. First, the variational autoencoder directly tries replicating its input by assigning probability measures on the latent layer between an encoder and a decoder network. On the other hand, the generative adversarial network is a competition between a generator network that takes a randomly generated latent vector to output a replica that a separate critic network evaluates as fake or real. The generator gradually learns to replicate the input data by adjusting its weights and biases to a random Gaussian noise input. The critic also learns to better distinguish fake from real. These adversarial networks have to balance the competition between the two neural networks. Balance and stability were obtained by applying the Wasserstein loss and gradient penalty.

In this project, four versions of each variational autoencoder (VAE) and Wasserstein generative adversarial network with gradient penalty (WGAN), only differing in the size of the latent dimension, are set up to produce synthetic populations. No other settings but the latent dimension vary between the different VAE or WGAN models. The choice of architecture is based on the experiences from the transporting research articles [6, 17] and initial runs to find one setup that fitted. No efforts are made to fine-tune or optimise the models, as this is premature at this stage of experimentation, where the general fitness of such methods is investigated. Unless such fitness is finally proven, there is no need for fine-tuning.

3.3.1 Variational Autoencoder Architecture

The variational autoencoder has a mirrored two-linear-layers configuration with 50 and 100 neurons in the decoder and 100 and 50 in the encoder. Experiments from transport research showed that more layers or nodes in the networks perform worse on the basic metrics of standardised root-squared mean error, Pearsons and R-squared. During the initial runs, this architecture performed well and was kept during the project. In addition, the beta-variational autoencoder configuration is used. To tweak training performance, it applies a constant beta to the Kullback-Leibler divergence loss. The kernel trick configuration, which applies the statistical metrics to the latent representation, uses Kullback-Leibler divergence (KL) loss which measures the similarity between probabilities. To measure construction loss, the difference between input and replica, the RMSprop as loss functions. KL measures the similarity between feature probabilities in the original and replica, while RMSProp measures the overall reconstruction loss. Code is found here A.1. An excellent mathematical explanation of variational autoencoders is found here [14]. The setup differs from the one used in transport research [6] only by applying sigmoid activation on all single outputs instead of using softmax. These are nevertheless completely equivalents as softmax is a clustered implementation of sigmoid activation.

The following settings are used for the variational autoencoder:

- beta: 0.5
- learning rate: 0.0001
- batch size: 115/156
- latent dimensions: 15, 30, 50, 100
- linear layers with batch normalisation
- activation: leaky relu 0.2
- output layer: sigmoid on each feature
- number of training epochs: chosen at approximate convergence for each dataset

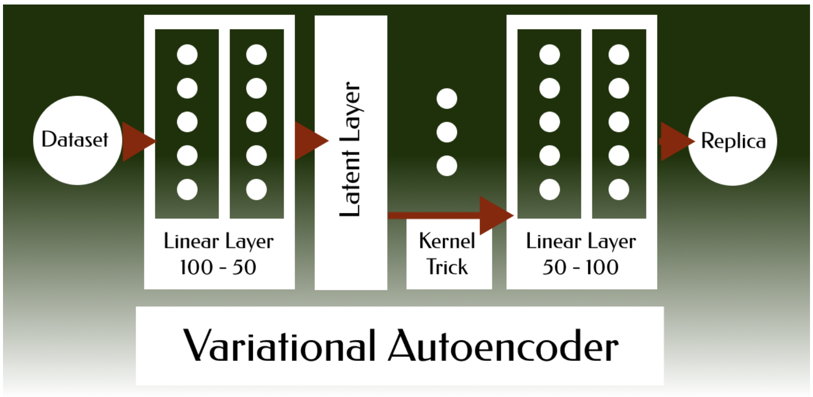


Figure 7: Architecture of the applied variational autoencoder used for Finland and Norway EU-SILC data.

3.3.2 Wasserstein Generative Adversarial Network Architecture

After testing the setups from the transport research article [17], the Wasserstein generative adversarial network with gradient penalty was finally chosen as the basic architecture. Code is found here B.1. The gradient penalty stabilises learning and ensures convergence better than the weight clipping in the referred article. Arjovsky [1] that managed to stabilise generative adversarial networks with Wasserstein loss, encouraged using gradient penalty to weight clipping, as his team experienced training that either generated poor sample or failed to converge with Wasserstein loss and weight clipping. Otherwise, the general architecture from the article [17] is used. The following general settings were applied:

- linear layers in generator: 150 and 100
- linear layers in critic: 100 and 150
- activation: leaky relu 0.2
- output layer generator: sigmoid for each feature
- output layer critic: one node linear layer with no activation
- learning rate: 0.0001
- batch size: 115/156
- critic iterations: 5
- optimiser Adam: beta (0.5, 0.9)
- lambda for gradient penalty: 10.0
- latent dimensions: 15, 30, 50, 100

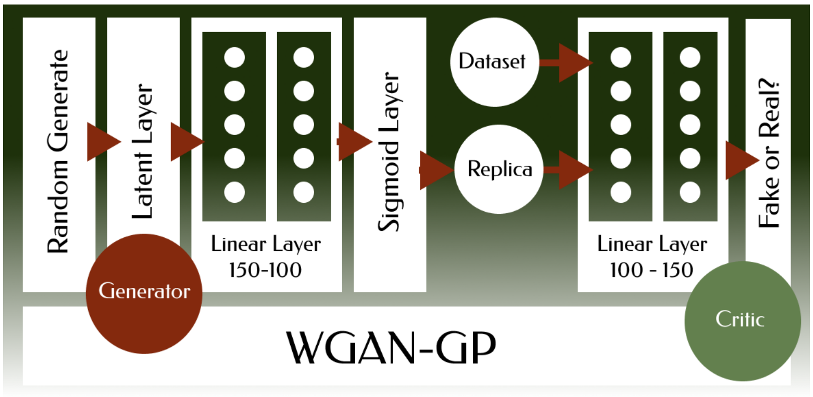


Figure 8: Architecture of the applied WGAN-GP used for Finland and Norway EU-SILC data.

- epochs training: adjusted to convergence for each dataset

3.3.3 Neural Manifold Clustering Embedding

Neural Manifold Clustering and Embedding (NMCE) [24] is a contrastive self-supervised method based on maximal coding rate reduction [49] combined with a similarity separator that adds augmented examples close to the original. In this project, the VAE model assists contrastive learning by augmenting records using the latent space from the original to a replica from the VAE decoder. Replica and original are the inputs to train the NMCE. Code is found here C.1. This combination of originals and replicas in learning is the operative constraints function in NMCE [24]. This clustering technique is preliminarily tested to compare the original and synthetic populations according to patterns.

3.4 Reliability and Validity

The literature on reliability and validation in experimental social sciences is concerned with outcomes and interpretation of overall results from the experiment [9], which is different from comparing a sufficient similarity between original and synthetic data. On the other hand, researchers in small-area estimation and population synthesis within agent-based simulations have dealt with the evaluation of low-attribute populations for decades but still need to reach a consensus on best practices [31, 22, 41, 30]. According to any definition of a synthetic population that encompasses similar statistical properties with inhabitants in a specific geolocation, it is evident that the statistical patterns in the geolocation must match those in the synthetic population. Good quality implies similarities in single variables and multi-correlations across all variables for low-attribute populations. This project uses the root mean squared error, standardised root mean squared error, Pearson's correlation coefficient, and R-squared to compare similarity. However, it is only feasible to assess similarity across every single variable and all binary correlations, as testing for all possible multi-correlations is intractable with over 150 attributes.

How, then, do a thorough evaluation of synthetic populations with high numbers of attributes? By acknowledging the embeddedness of synthetic populations in this project to public health and policy analysis and planning, Cook and Campbell's suggestions for validating quasi-experiments can bring light to some of the challenges. The assumption is that the synthetic populations proposed in this work should answer what-if questions about health outcomes resulting from an intervention. By exploring health outputs from different potential interventions, policy planning for reducing inequality in health can take place. The synthetic population, then, needs to function as an equivalent to the original data in such analysis. From these assumptions, the validation of the synthetic populations can proceed from the general similarity measures sketched above.

Cook and Campbell [9] suggest four categories of validity deriving from the concepts of internal and external validity suggested by Campbell and Stanley in the early 1960-ies. Cook and Campbell define validity as "...the best available approximation to the truth or falsity of propositions, including propositions about a cause." [9, p. 37]. Cook and Campbell apply a critical-realist approach to cause founded in, among others, the theories of John Steward Mills's inductive approach and Carl Popper's falsification [9].

3.4.1 Statistical Validity

Statistical conclusion validity is the first category with significant concerns regarding falsely concluding with strong covariance when it does not exist (type I errors) and incorrectly stating no difference when it is a factual difference (type II errors). In addition, statistical power and magnitude of change with appropriate confidence intervals are also at stake in this category. The metrics suggested for evaluating synthetic populations above belong to this category, as they analyse variance and errors in variance. These measures are partly robust to violations of normality, implying that variables with slightly off-pist distribution from Gaussian, nevertheless, can be handled well. However, any uncorrelated errors caused by the generation techniques will make these metrics error-prone [9, p. 42]. Two conditions of threats to statistical conclusion validity at stake for synthetic populations are the reliability of measures and the random heterogeneity of respondents. Variables that reproduce low scores on metrics like root mean squared error, Pearson's and R-squared, by definition, have low reliability compared to the original data and cannot necessarily be trusted when used in an experiment. This unreliability adds to any reliability issues in the original data, a question that is irrelevant to this project but highly relevant in an epidemiological interpretation of results. The second condition concerning random heterogeneity of respondents is crucial as some generative methods push variable values to the mean and produce fewer examples with extreme values than are represented in the original data. When this happens, the statistical conclusion validation is threatened, which is particularly challenging in this project, where the synthetic population is used to analyse vulnerable groups perhaps identified by some extreme values. The synthetic population will be biased towards the mean population, with less power to identify subgroups of interest.

A short comment on reliability is that when assessing the techniques of producing the synthetic populations, these techniques are fully reproducible and hence reliable, given the use of the same code, tools and random seeds. The lack of reliability above concerns the need for more equivalence between the original variable values and the synthetic population. Therefore, reliable and valid terms are multifaceted, as synthetic populations are evaluated on multiple levels.

The following metrics are reported for evaluating a synthetic population [22, 48]:

- TAE (total absolute error) and RAE (relative absolute error) measuring similarity in marginals distributions
- AAPD (absolute average percentage difference)
- SRMSE (standard root mean squared error) or other squared or root squared measures for error to evaluate the goodness of fit between original and synthetic marginals
- Pearson's correlation coefficient measures the strength of the relationship between the original and synthetic data point, with one being a perfect match and zero indicating no correlation.
- R-squared is the coefficient for determination representing the variation in the data, that is, how well the data fit the regression line with one indicating a perfect match, and zero that the proposed model does not explain anything.
- Z-scores and standard deviation
- Proportion of good predictions (PGP) measuring the proportion of misclassified entities
- KL (Kullback-Leibler divergence) measuring similarities between two probabilities
- Cramer's V, Pearson's correlation coefficient, and R Squared (R2) measuring strength in dependencies
- Bland-Altman to visually and statistically (by mean and confidence intervals) compare two instruments of measurements of the same variables
- Comparing correlation between pairs or multiples of variables

3.4.2 Internal Validity

Internal validity relates to the causal models and the conclusions drawn from results [9]. The critical takeaway for synthetic populations is the threat to internal validity caused by a shift in the scales of the metrics, called instrumentation. If we assume the original data set the scale, any truncations of that scale by, i.e. fewer examples in the extreme high or low categories, is such a threat. In this project, all numerical, ordinal or nominal categorical variables are one-hot-encoded, avoiding scaling challenges. However, as mentioned in the section on statistical validity above, such truncations are challenging and need to be handled.

3.4.3 Construct Validity

Construct validity is about operationalising parts in the problem to be investigated by an experiment [9]. There are no relevant threats under this heading. However, then Cook and Campbell propose to create so-called nomological nets showing predicted patterns of relationships that would permit having the chosen construct guiding an experiment. The outputs from the heterogeneous treatment effect by Causal Random Forest DML on original data can function as such nomological net, being the reference for outputs using synthetic data.

3.4.4 External Validity

External validity is a question of the generalisability of results to other areas or across groups [9]. For example, suppose the produced synthetic populations were used in a particular geographical region to guide interventions to deal with inequalities in health; given the original data as ground truth, the synthetic populations needed to possess similarities of heterogeneity guaranteed by the original data's proper random sampling. As mentioned above, if the deep generative method pushes examples to the mean or otherwise hampers the reconstruction of extreme values, this also threatens external validity. Deliberate heterogeneity sampling can mitigate this threat [9].

3.5 Causal Forest as Quasi-Experimental Evaluation Tool

The machine learning tool causal forests offer a data-driven approach to identify and measure heterogeneous treatment effects from observational data as described by Athey and Wager [2] and have been recently successfully applied to re-analyse well-conducted randomised control trials that resulted in negative findings (refs), to reveal subgroups with opposite outcomes that nulled out the effect for each other in the original trial. Causal Forest allows any type of forest, including classification and regression forest, for provably valid statistical inference and is based on an asymptotic Gaussian and centred sampling distribution. The conceptual and mathematical details are available in this article by Athey and Wager [2].

The Causal Forest, described by Athey and Wager [2], generates regression trees by using some data to set the parameters and assign other data to the tree's leaves. This division of data usage is called "honesty" and makes the model robust and less biased [7], and safeguards valid point estimates and confidence intervals. Unfortunately, the division trick wastes half of the data. Nevertheless, the method has proven highly effective and precise [2]. Moreover, while a random forest technique only assigns samples, a causal forest calculates the leaves samples' mean treatment effect and confidence intervals. Causal forest methods can therefore be helpful in any microsimulation with a quasi-experimental design, as they, by computing valid point estimates and confidence intervals, partly solve the problem of uncertainty estimation [36] [43].

A significant challenge for governance in public health is to target interventions to decrease inequalities in health [20]. Causal forests can help inform customised interventions to reach these goals. Furthermore, because Causal Forest identifies and evaluates complex statistical properties in the data, it can also be a tool for externally validating synthetic populations.

Based on the quasi-experiment design in this project, the evaluation is done by running original and synthetic data through a Causal Forest model trained on original data only to predict health outcomes as self-perceived health (PH010). Five different interventions (treatments) are selected. The first and second are two measures of education (PE040). The third is meeting with friends, a social metric (PD050). The fourth is leisure activities (PD060), and the fifth is the degree of nervousness as a psychological factor (PW050). These experiments are illustrative but represent the statistical dependencies without carving out natural and causal relations. The metrics used for evaluating outputs from the Causal Forest method are the CATE (conditional average treatment effect). The results are visualised as a graph tree with three or four layers and Shap variable importance plots.

3.6 Policy Game on Public Health as Context

The following figures 9, 10, 11, 12 show a preliminary setup of a dashboard public health policy game in the making to demonstrate usage of synthetic population data in policy analysis and planning with suggestions for how to simulate policy scenarios. The public health arena, dealing with inequalities in health, is the broad context for evaluating synthetic populations. Embedding this broader context into a policy game is framing the context for the following evaluation of synthetic populations.

A short walk-through of the policy game in the making shows that a main dashboard in figure 9 displays the content of the synthetic population. Next, the public health profile in figure 10 shows scores on various selected variables for a particular region compared to the mean for all regions or a single, i.e. best-practice region. Next, the policy dashboard in figure 11 offers the player an exploration of

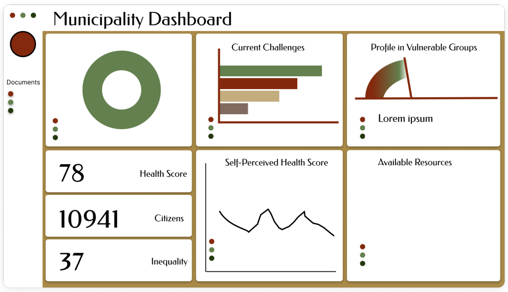


Figure 9: Policy Game Main Dashboard

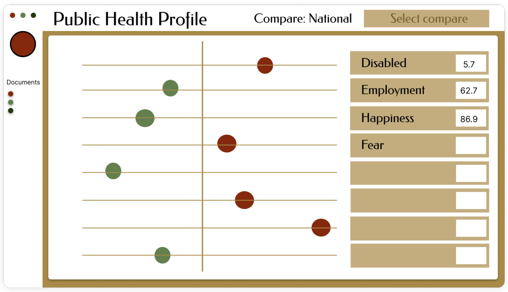


Figure 10: Public Health Profile

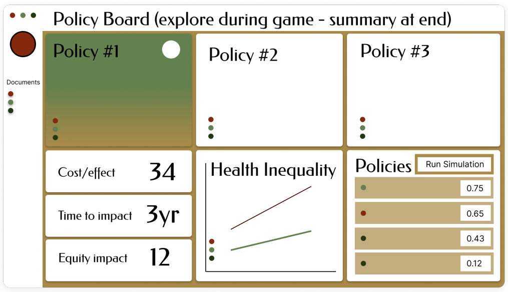


Figure 11: Policy Dashboard

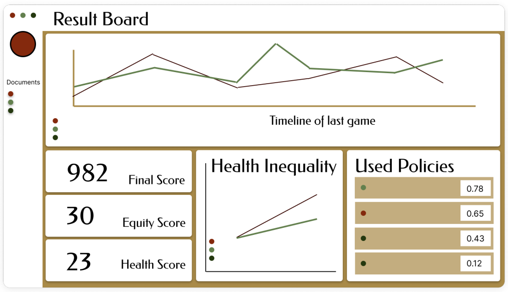


Figure 12: Result Board

different interventions to assess their potential impact on vulnerable groups and eventually offers choices for implementation. Finally, the result board in figure 12 shows the results of an intervention by, i.e. drawing data from a simulated new synthetic population.

These sketches suggest some general aspects of using synthetic populations in a policy game. The main point of suggesting a policy game model as a contextual embedding for synthetic populations is that the setup, to some degree, mimics the surveillance and work needed by governmental bodies to deal with inequalities in health, and next, because it is a simplified representation of a complex field of operation that can aid learning. The learning potential is motivated in its own right. Evaluating the synthetic populations against epidemiological standards serve two purposes. First, it is used to assess authenticity, and second, it opens a transition to an ambition to raise the standards of synthetic populations to fit real-life epidemiological analysis.

3.7 Tools

PyTorch 2.0 is the tool to set up and run deep generative models for population synthesis and clustering techniques. While Tensorflow and Keras can also be used, these tools turned out less flexible and transparent than PyTorch, especially when saving and reusing customised models with submodels like the variational autoencoder and generative adversarial network. Pandas version 1.5.3 are used for original preprocessing data in a CSV format (EU-SILC Finland) or SPSS format (EU-SILC

Norway). The EU-SILC data was imputed using sklearn.impute.IterativeImputer from Scikit-Learn version 1.1.3. EconML and its model CausalForestDML version 0.13.1 are used out of the box on the observational data to generate heterogeneous treatment effects. All code is mainly run in Jupyter-Notbook version 6.5.4 on either a MacBookPro (without GPU) or Linux (Ubuntu 22.04 LTS) (with GPU). Visuals are created using the Seaborn package version 0.12.2, mainly based on Matplotlib-Pyplot version 3.7.1. Customised code is written to calculate, i.e. SRMSE, while Pearson's correlation coefficient and R-square are calculated from Numpy version 1.23.5 and Statsmodel version 0.13.5.

3.8 Experimental Design

The design to answer the research questions is built up by first generating four different versions of each of the two generative models with latent representation dimensions of 15, 30, 50 and 100. Next, these eight models are run on the original EU-SILC datasets for Finland and Norway, respectively, resulting in sixteen synthetic populations.

These populations are assessed according to root mean squared error, Pearson's correlation coefficient, and R-squared for all single one-hot-encoded attributes and all binary combinations of those attributes. These results represent the first line of answering RQ1 and RQ3b and cover issues related to statistical validity, the creation of possible uncorrelated errors, and the level of reliability for each variable. These results also contribute to answering the threats to internal validity linked to the level of truncation giving ceiling and basement effects.

Next, these synthetic populations are compared with original data in Bland-Atman analysis, where the original data are a baseline measurement to which the synthetic populations are compared. The variables falling outside the confidence intervals and the confidence intervals themselves contribute to answering RQ1 and RQ3b by informing the reliability discussion under statistical validity. This metric cannot identify the causes of unreliable reproductions of variables. However, it does show that if variables fall out of the confidence intervals, there is a chance for uncorrelated errors, as there are 95 per cent likelihood that these variables have a pattern that does not, by random, fit with the original data.

Two other approaches are made to look for hidden differences between the original and the synthetic populations to determine how fit they are for health outcome analysis, which is the cornerstone of policy analysis (RQ1) and epidemiological compatibility (RQ3). These are contingency matrices between the health outcome variable self-perceived health (PH010) and other variables known to impact health inequalities, like education (PE040). Next, confusion matrices are produced to look for a more detailed spreading of individuals between the original and synthetic data. For simplicity, only the VAE-50/100 and WGAN-50/100 from the Norwegian data are used for this analysis. It is expected, in principle, to be similar to using other models or the Finnish data. Results from accuracy, precision and recall in the visual contingency and confusion matrices contribute to answers RQ1 and RQ3, particularly by enlightening the internal validity question of truncation and the external validity question of underrepresenting outlier examples. These matrices can also function as the so-called nomological nets described under construct validity.

RQ2 is approached by looking at the reproduction of regions in the Norwegian dataset and running a direct numerical upscaling by the generative models while measuring changes in accuracy, precision, recall and the balance between precision and recall, F1. In addition, cluster patterns from the neural manifold and embedding [24] are compared across the regions in original and synthetic populations. However, as neural manifold clustering and embedding is a novel technique derived from imaging, not yet used on tabular data, it is only preliminarily suggestive as a measure of similarity for synthetic populations.

RQ3a can be transformed into a question of authenticity that can directly be answered by RQ3b. However, other options relevant to educational purposes are more than just the synthetic population meeting the strict requirements under RQ3b. These questions will be dealt with in the discussion.

RQ3b is approached by looking to which degree the synthetic populations output similar results to the original data in running quasi-experiments using Causal Forest DML. The output from this analysis is trees pinpointing red or green colours when a treatment has a negative or positive effect on the current fraction of examples in the leaf nodes. Additional outputs are Shap diagrams from the causal forest, showing a ranked list of the variables having the most significant impact of treatment on the outcome variable self-perceived health. The similarity in ranking and direction between the original data and the synthetic indicates that some basic correlations, causal or not, are reproduced.

4 Results

4.1 Synthetic Populations by Deep Generative Methods

Synthetic populations are produced by the variational autoencoder [6] and generative adversarial network with Wasserstein [17], using different sizes of 15, 30, 50 and 100 for the latent representation of EU-SILC for Finland (230 one-hot-encoded features) and Norway (156 one-hot-encoded features).

The univariate differences between one-hot-encoded variables in the original and synthetic populations are visually presented with metrics for standardised root mean squared error, Pearson's correlation coefficient, and R-squared for EU-SILC Finland and Norway. These metrics show differences between the marginals, the mean of a variable representing the variable's probability of appearing over the complete set of features, between marginals in original versus synthetic populations. A lower standardised root mean squared error implies less linear distance between the original variable and the one produced by the deep generative method. A complete match is indicated when Pearson's correlation coefficient and R-square equal one.

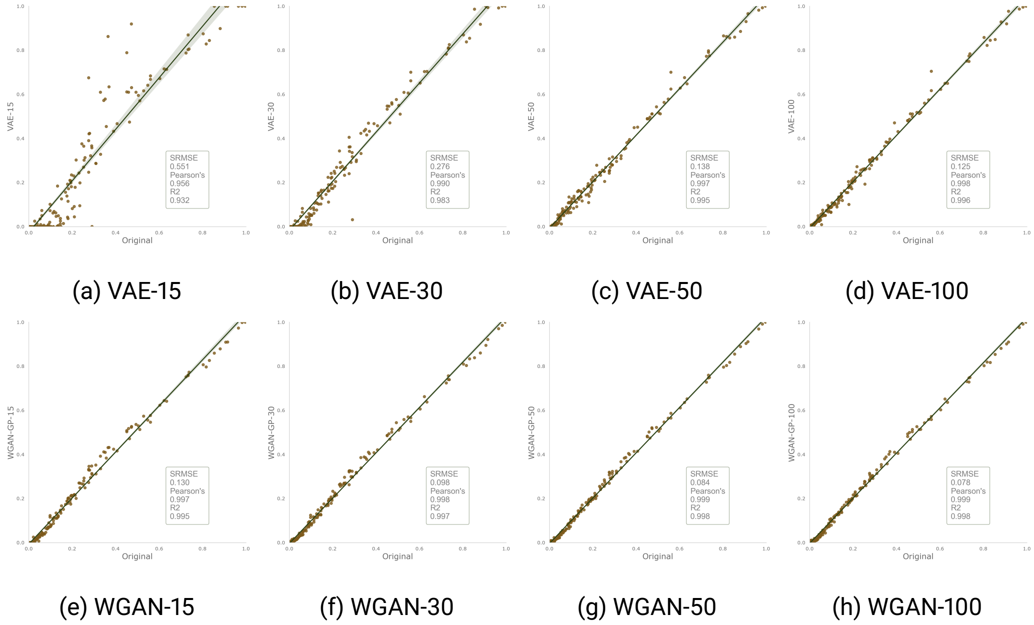
The similarity in the binary correlations between all variables within each dataset is visualised using root mean squared error, as the input marginals are normalised by multiplying their marginals over the complete set with each other. Next, the two Pearson's and R-square correlation measures are shown in the figures 16.

Similarity measured by the standardised root mean squared, Pearson's correlation coefficient, and R-squared improved gradually with increased size for the latent space from 15 to 100 for both deep generative models. The Wasserstein generative adversarial network generates populations and single variables more similar to the original than the variational autoencoder for both datasets as shown in univariate correlation 13 14, bivariate correlations 15 16, Bland-Altman 20 and single variables E F.

The Bland-Altman plots, suggested as metrics for synthetic populations validation by Yamego [48], visualise the original and synthetic data as two different methods to measure features, with the original data as the baseline. These plots show a scatter plot for features with confidence intervals capturing variables being random within the borders and variables differing more outside. For example, results from the Bland-Altman show that the VAE population have more variable outside the confidence intervals than the WGAN population and that increased latent dimension narrows confidence intervals for all models, as shown in Figure 20.

A test where the original data and the synthetic populations are split up into clusters generated by training the neural manifold clustering and embedding algorithm on original data shows that only the largest cluster maintains the same results on standardised mean squared error, Pearsons and R-squared. All other clusters with more than 200 examples showed worse similarity measured by this method than their original counterpart on the Norwegian dataset 4.

The reproduction quality of each variable, as shown in E, gets better with a higher latent dimension. However, some variables still reproduce worse and do not keep within the confidence limits in the Bland-Altman plot. This threatens the statistical and internal validity 3.4. A variable that contains to be poorly reproduced is ”work” (SRMSE 0.413 in VAE-50 and 0.339 in VAE-100) in the EU-SILC Norway and PB190, marital status (SRMSE 0.204 in VAE-50 and 0.339 in VAE-100). The results are better for similar WGAN models, but ”work” still has the most significant error compared to other variables. The WGAN models handle marital status better. ”work”, and marital status (PB190) are nominal, with work having eleven categories and marital status 5. Of the other variables with five or more categories, these are the only nominal ones except for region, type of housing and economy. Similar patterns are found in the Finnish dataset where PL031 represent ”work”, and PB190 is like in



(a) VAE-15 (b) VAE-30 (c) VAE-50 (d) VAE-100

(e) WGAN-15 (f) WGAN-30 (g) WGAN-50 (h) WGAN-100

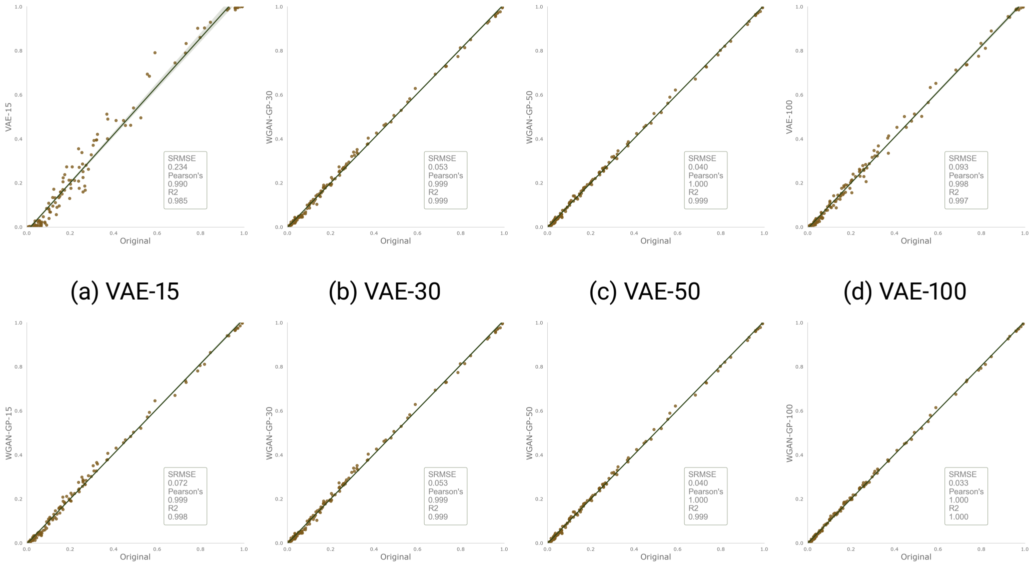
Figure 13: Match between single variables in original and synthetic data from variational autoencoder and WGAN. Comparison is made on 230 one-hot-encoded and binary categorical variables from EU-SILC Finland.

the Norwegian dataset. Regions are not a variable in the prepared Finnish dataset.

All models' architecture is presented in the method chapter 3, and the code is available in the Appendix A.1 B.1 C.1. Each generative model was run to convergence. A sample of learning curves from training the Wasserstein generative adversarial network on the EU-SILC Norway is available in Appendix 39. Each variable in all combinations of models and datasets is visualised in plots with original besides synthetic split on gender ("isFemale"), with standardised root mean squared error, Pearsons and R-squared for EU-SILC Finland and Norway in the Appendix E. The split on gender allows for a visual inspection of the bivariate correlations with "isFemale" while showing the univariate correlations between original and synthetic data.

4.2 Scaling of Synthetic Populations

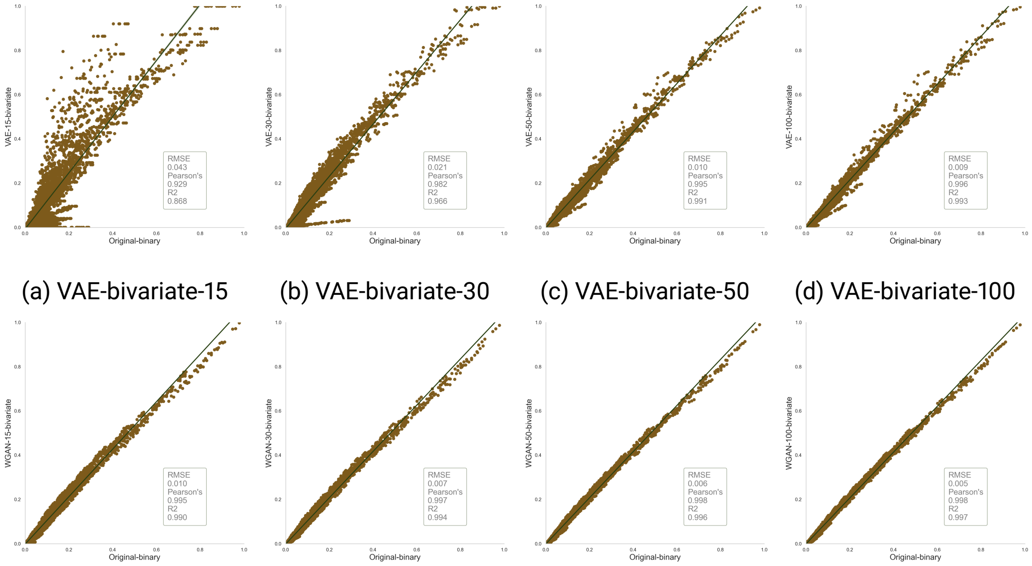
A non-scaled synthetic population is a replicas population similar to the original data's number of examples. All previous tests are run on such populations. In this section, the scaling of synthetic populations is investigated. The most straightforward approach is to reproduce more replicas. However, as noted above in the results of general deep generative synthesis, some drawbacks of truncating and non-random variables identified in the contingency and confusion matricesD and the Bland-Altman plots 20 need to be considered. An alternative to the simple upscaling above is to train



(a) VAE-15 (b) VAE-30 (c) VAE-50 (d) VAE-100

(e) WGAN-15 (f) WGAN-30 (g) WGAN-50 (h) WGAN-100

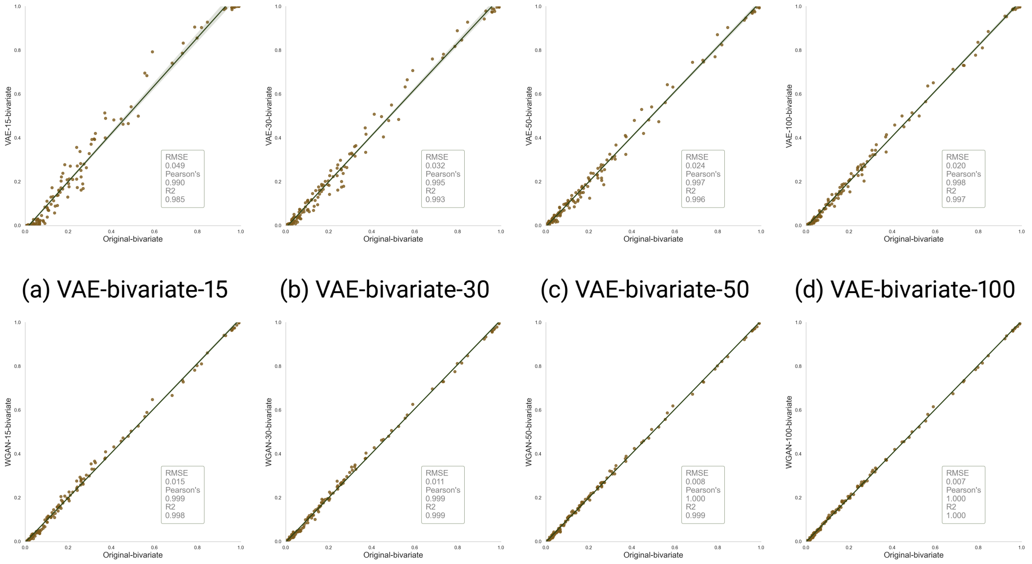
Figure 14: Match between single variables in original and synthetic data from variational autoencoder and WGAN. Comparison is made on 156 one-hot-encoded and binary categorical variables from EU-SILC Norway.



(a) VAE-bivariate-15 (b) VAE-bivariate-30 (c) VAE-bivariate-50 (d) VAE-bivariate-100

(e) WGAN-bivariate-15 (f) WGAN-bivariate-30 (g) WGAN-bivariate-50 (h) WGAN-bivariate-100

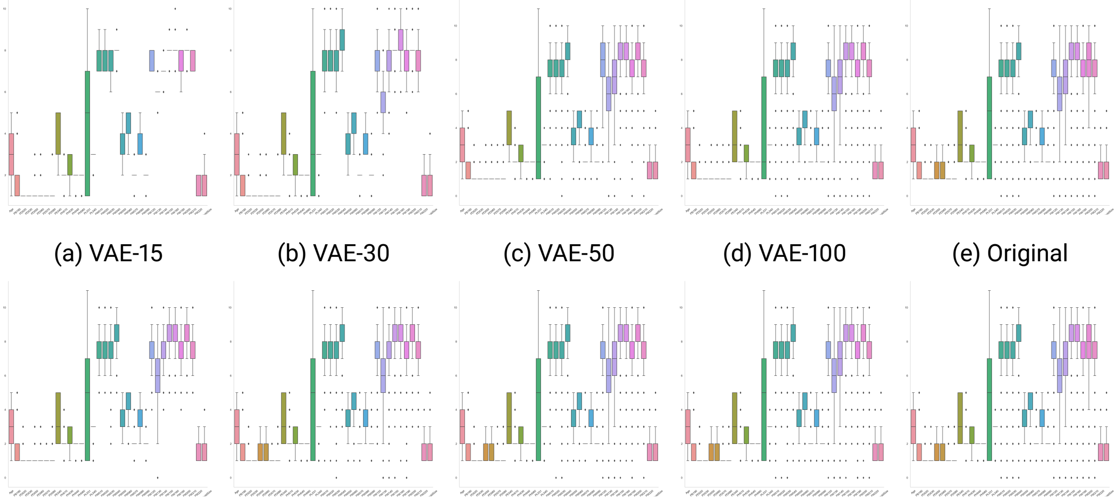
Figure 15: Match between all pairs of variables in original and synthetic data from the deep generative method. Comparison is made on all pairs of 156 one-hot-encoded and binary categorical variables from EU-SILC Finland.



(a) VAE-bivariate-15 (b) VAE-bivariate-30 (c) VAE-bivariate-50 (d) VAE-bivariate-100

(e) WGAN-bivariate-15 (f) WGAN-bivariate-30 (g) WGAN-bivariate-50 (h) WGAN-bivariate-100

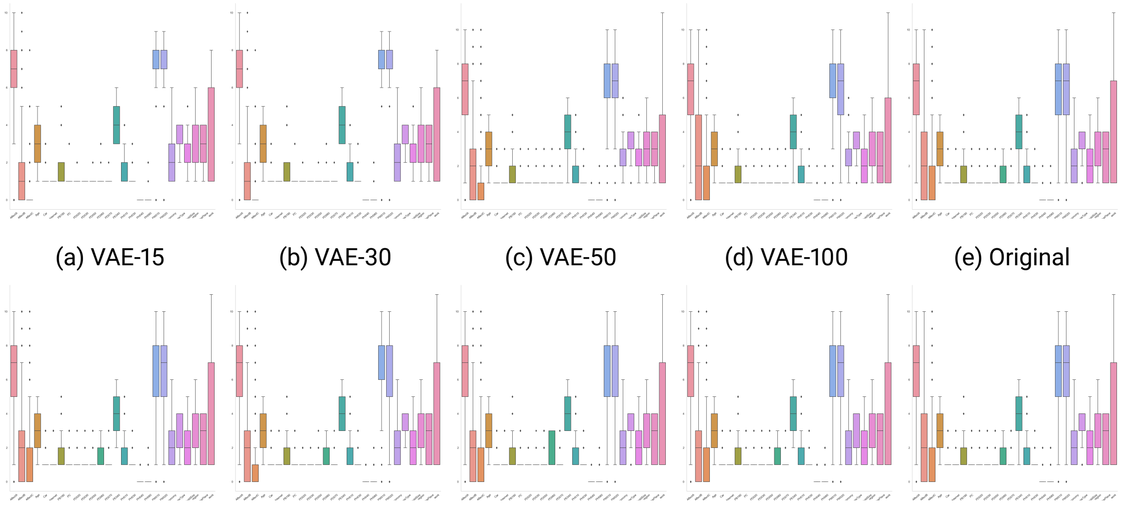
Figure 16: Match between all pairs of variables in original and synthetic data from the deep generative method. Comparison is made on all pairs of 156 one-hot-encoded and binary categorical variables from EU-SILC Norway.



(a) VAE-15 (b) VAE-30 (c) VAE-50 (d) VAE-100 (e) Original

(f) WGAN-15 (g) WGAN-30 (h) WGAN-50 (i) WGAN-100 (j) Original

Figure 17: Box plots of variables except for binary from original EU-SILC Finland and all generative models. The one-hot-encoded feature is transformed back to its original variable categories for displaying. The original data are plotted at the end of each line for visual readability. The upper and lower lines in each plot are the confidence intervals for each variable. The coloured boxes divide at the median; the top is the 75 percentile, while the bottom is the 25 percentile.



(a) VAE-15 (b) VAE-30 (c) VAE-50 (d) VAE-100 (e) Original

(f) WGAN-15 (g) WGAN-30 (h) WGAN-50 (i) WGAN-100 (j) Original

Figure 18: Box plots of variables except binary from original EU-SILC Norway and all generative models. The one-hot-encoded feature is transformed back to its original variable categories for displaying. The original data are plotted at the end of each line for visual readability. The upper and lower lines in each plot are the confidence intervals for each variable. The coloured boxes divide at the median; the top is the 75 percentile, while the bottom is the 25 percentile.

models to produce, i.e. regional populations. In the Norwegian dataset, the smallest Region Three has 1632 (6.7 per cent) examples. This Region is the worse of all regarding RSME, Pearson's and R-squared and is also reproduced with fewer examples. The best variational autoencoder model outputs only 974 (4 per cent) examples for Region Three. The assigned examples are visualised for all VAE and WGAN models in contingency tables here 21.

Note that when doing simple upscaling, the least affected is Region Three. Upscaling by dividing the six regions could solve the generally lousy reconstruction. Two approaches were tried. The first was to train an autoencoder on Region Three only and later replace the examples matching Region Three with the newly generated ones. This attempt was quickly abandoned because, even though Region Three got a better match, the total population could have done better. The second alternative was training all regions separately on variational autoencoder with latent dimension 100 and then merging them into one population. The match for each Region was almost as good as training one model on the total population with SRMSE at 0.106 (against 0.093) and Pearsons at 0.998 (against 0.998), and R-squared at 0.97 (against 0.997). The bivariate correlations also give similar results, but in this case, the RMSE is 0.012 against 0.007, the Pearsons 0.997 against 1.0, and the R-squared 0.994 against 1.0, indicating that the merged model is slightly worse at reproducing binary correlations. Results, including the Bland-Altman plot, are shown in figure 22. The ordinary

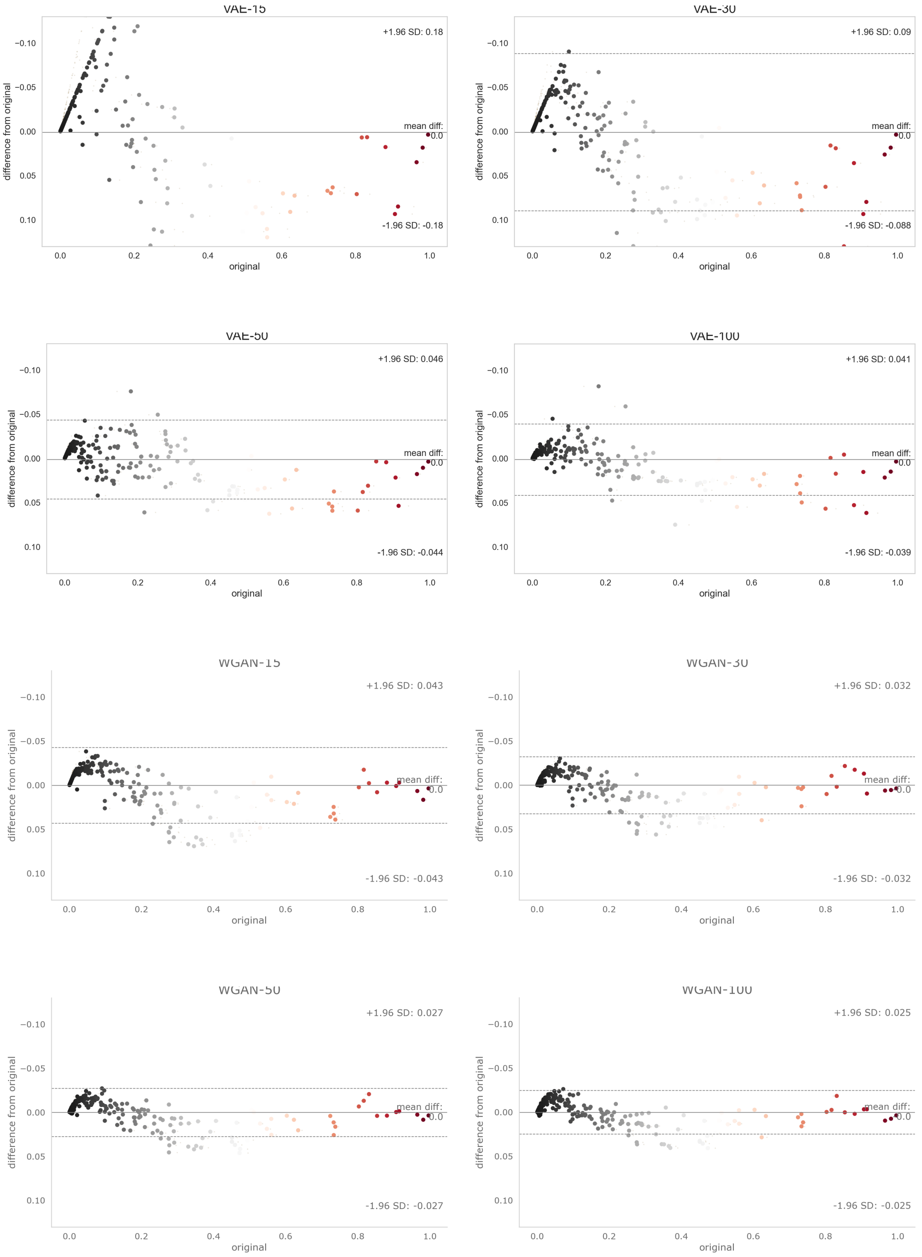


Figure 19: Bland-Altman for populations generated by WGAN and VAE. Each figure compares against the original data for 230 one-hot-encoded features from EU-SILC Finland. The dotted lines are upper and lower confidence intervals at 95 per cent.

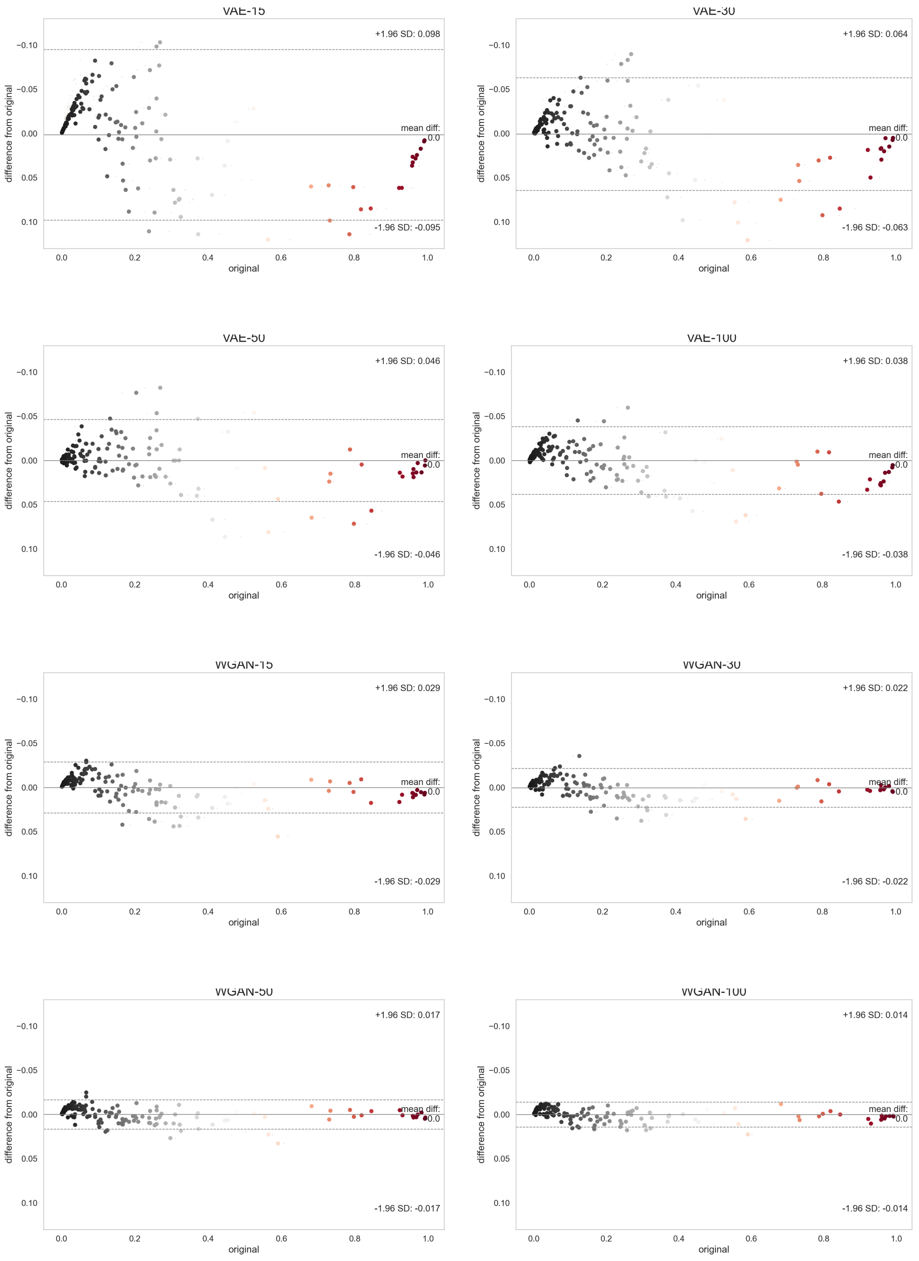


Figure 20: Bland-Altman for populations generated by WGAN and VAE. Each figure compares against the original data for 156 one-hot-encoded features from EU-SILC Norway. The dotted lines are upper and lower confidence intervals at 95 per cent.

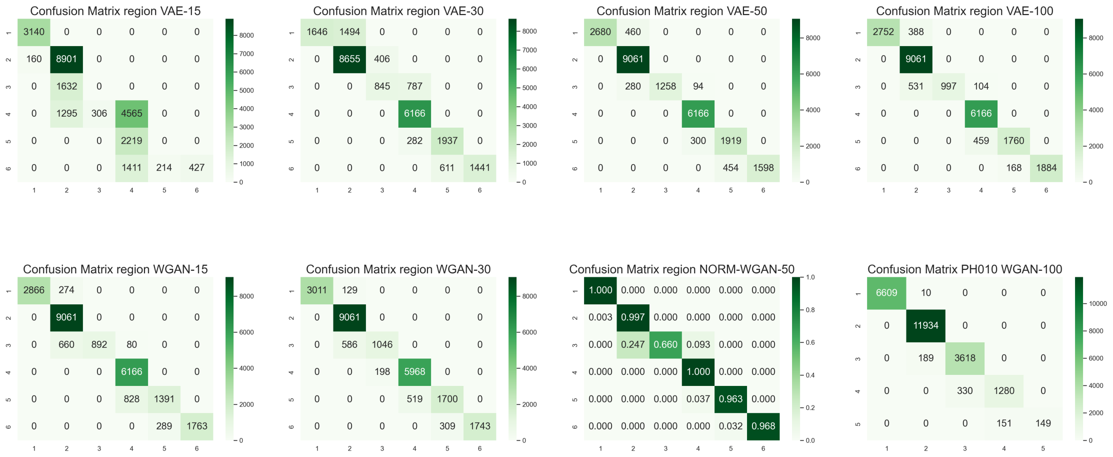


Figure 21: Confusion matrices for all VAE and WGAN models for regions in the EU-SILC Norway dataset. The comparison is made by sorting the original and synthetic datasets of similar size on Region and then plotting the hits in original versus synthetic datasets.

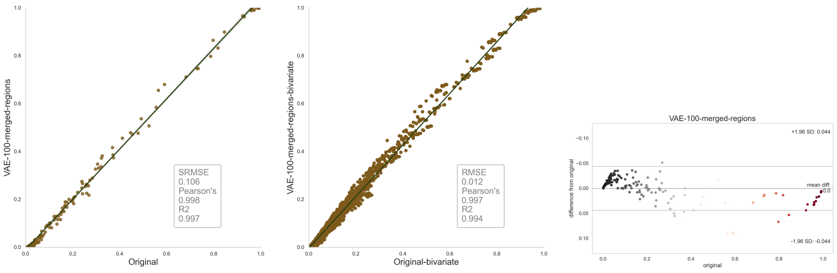


Figure 22: Univariate and bivariate plot with Bland-Altman for regions produced separately with VAE-100 architecture and next merged to a complete population on the EU-SILC Norway.

VAE-100 trained on the entire population is shown in figure 14d 16, and its Bland-Altman plot is shown in figure 20.

A test on the VAE-50 and WGAN-100 showed that the precision is guaranteed. There is no change in standardised root mean squared error, Pearson's correlation coefficient, or R-squared when measuring upscaled populations' means against the original data. However, the recall gets steadily worse, accompanied by a steady rise of F1 when upscaling, indicating that those attributes that are worse represented in the unscaled version do not unexpectedly get even more badly represented. These are the exact effects of misclassification shown in the contingency and confusion matrices D that are magnified when upscaling.

A test on the relationship between the output variable self-perceived health (PH010) and education (PE040) in the original data in figure C and the same relationship from synthetic data figures C shows the tendency to underrepresent combinations that are more sparsely represented in favour of overrepresenting those better-populated alternatives. Furthermore, these examples have no scaling, which means the synthetic population is generated with the same number of examples as the original data.

Clusters	VAE-50	WGAN-100	c-VAE-50	c-VAE-50-aug
0	0.112/0.997/0.996	0.033/1.000/1.000	0.118/0.996/0.995	0.104/0.997/0.996
1	0.246/0.986/0.981	0.153/0.995/0.993	0.164/0.996/0.994	0.151/0.996/0.994
2	0.355/0.975/0.965	0.210/0.991/0.988	0.170/0.996/0.995	0.167/0.995/0.993
3	0.324/0.977/0.968	0.215/0.99/0.986	0.265/0.990/0.984	0.226/0.992/0.988
4	0.299/0.980/0.972	0.188/0.992/0.989	0.204/0.994/0.991	0.203/0.993/0.990
5	0.324/0.979/0.970	0.216/0.991/0.987	0.179/0.995/0.993	0.214/0.992/0.989
6	0.221/0.990/0.986	0.215/0.991/0.987	0.202/0.994/0.991	0.225/0.992/0.988
7	0.326/0.976/0.967	0.257/0.985/0.980	0.236/0.992/0.987	0.218/0.992/0.988
8	0.432/0.960/0.0945	0.233/0.989/0.984	0.199/0.995/0.992	0.173/0.995/0.993

Table 4: Performance (SRMSE/Pearsons/R2) of variational autoencoders (VAE) and Wasserstein generative adversarial networks (WGAN) with different latent dimensions on the eight highest ranked clusters from neural manifold clustering and embedding (NMCE) on Norwegian EU-SILC data with 156 one-hot-encoded variables. The measures compare synthetic clusters with the respective original data clusters.

4.3 Quasi-Experiment with Causal Forest

Quasi-experiments using EU-SILC observational data from Norway are run on four toy interventions. The interventions are chosen from available variables in the EU-SILC, using self-perceived health as an outcome. The first experiment is to see which groups will improve or take harm regarding self-perceived health (PH010) by getting a higher education using variable PE040 as treatment in a heterogeneous treatment analysis. The second experiment is to improve the capacity to meet friends (”social”). The third is to improve the capacity for leisure activities (”leisure”), and lastly, to reduce the level of anxiety (”affect”). The results are presented as a three and four-level tree, splitting into the essential attributes, leading to subgroups that benefit or do not from the particular intervention (treatment). No effort is made to make sure that all the covariates, that is, all variables not being the outcome or treatment variables, actually are epidemiologically relevant. Nor are the experiments meant to be anything other than an illustration of how synthetic populations can be evaluated, but a setting for use in policy analysis in a game.

A qualitative assessment of the three levelled charts is done. Four-levelled charts are produced, but it is already at the third level. The synthetic populations fail to reproduce the output from the original data completely. For the interested reader, both three-level and four-level charts run on all synthetic populations from VAE-15 to VAE-100 and from WGAN-15 to WGAN-100, including the merged regions data run, are provided in the appendix 6. The interventions with education were run in three different modes, each with a different split on the level of education (the selected split at below four was marked zero, else one). As these were reasonably similar, only education B is presented here. All treatments were transferred to a binary of, i.e. higher education or not, decided on the selected split. Self-perceived health was also made binary, with those reporting one or two on the five-category scale having "good health" marked with a one for the outcome variable. All below score zero on the outcome variable. The split values for leisure (PD060) were one if below two and else zero, indicating a one for people affording leisure activities. The split value for social encounters was set similarly to leisure (PD050). The effect (AffectB) limit was set to four. Any below limit gets zero if an equal or higher one is set.

The charts are checked for reproducing the correct variables at each level. First level: The VAE-15 and WGAN-15 fail for all but the leisure intervention. The VAE-30 and WGAN-30 fail to correct the first split for all but the affect intervention. All models fail to get the first split for the social intervention correctly. However, all but VAE-15 and WGAN-15 get one of the original data second split variables as their first split. The experiment with the worse result is social gathering intervention; the best is on affect. For the effect experiment, all models except VAE-15 and WGAN-15 get the first two layers correctly while missing only one variable at the third layer. In the effect experiment, the original data highlights one strongly affirmative (green box) and one strongly negative (red box). When reproducing these boxes, the VAE-100 and WGAN-100 come close—the original data results in 9621 in the green box and 225 in the red. VAE-100 and WGAN-100 both produce 9561 and 160, respectively. WGAN-50 is the closest to getting the red box correctly, with 214 examples. However, WGAN-50 overestimates the green box by over 400 more examples than the original. Even if the split on variables is not the same, similar patterns can lead to an approximately similar result if the variables split upon are either close to each other like neighbouring categories in the same variable or the variables are involved in splits in the layer above or below the original. In the case of both epidemiological rigour and authenticity in a policy game, some equivalent but different patterns can

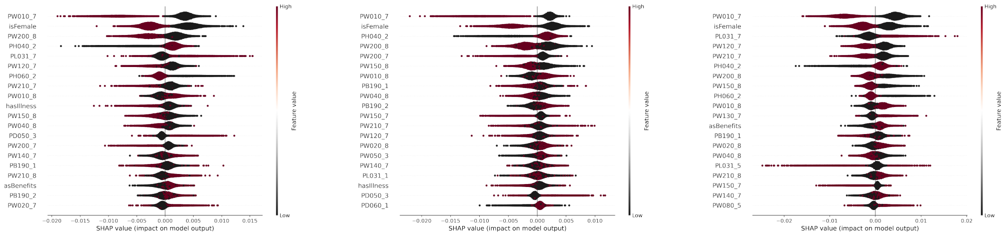


Figure 23: Original-Edu-A

Figure 24: WGAN-50-Edu-A

Figure 25: VAE-50-Edu-A

Figure 26: Variable importance in ranked order for original, WGAN-50 and VAE-50 populations from EU-SILC Norway on treatment Education A from Causal Forest. The outcome is self-perceived health (PH010).

occur without severe threats to validity.

Synthetic populations generated by WGAN score better on the general similarities measures like SRMSE, Pearsons and R-square than similar latent layer-sized VAE. However, the difference is hard to spot and not noticeable when inspecting the outcomes from heterogeneous treatment effects in the charts, where VAE-100 and WGAN-100 produce comparable results. Therefore, a second way of inspecting the results from heterogeneous treatment effects from Causal Forest DML is to look at the Shap plots showing the ranked order of covariate variables (those not outcome or treatment variables) and if their effect is to reduce or increase the score on the outcome variable. Results from these plots are placed in Appendix 6.

Comparing ranked variables of importance shows that the VAE population, more often than their counterpart WGAN models fail to rank the first variables correctly. Also, on this metric, the WGAN population is closer to the original. The WGAN has a reasonably good reproduction rate if comparing a rank equal to or one position from original data. Results from treatment Education A are shown in figure 26 and from Education B in figure 30.

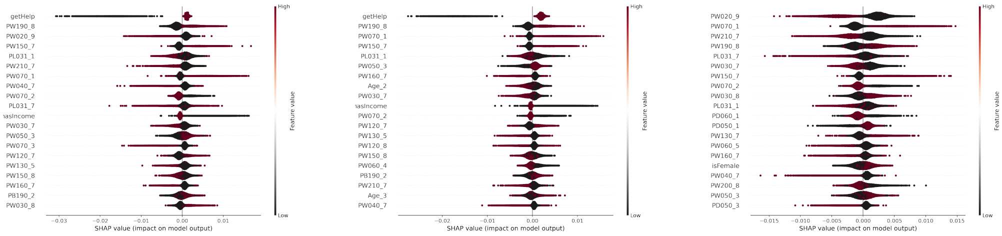


Figure 27: Original-Edu-B Figure 28: WGAN-50-Edu-B Figure 29: VAE-50-Edu-B

Figure 30: Variable importance in ranked order for original, WGAN-50 and VAE-50 populations from EU-SILC Norway on treatment Education B from Causal Forest. The outcome is self-perceived health (PH010).

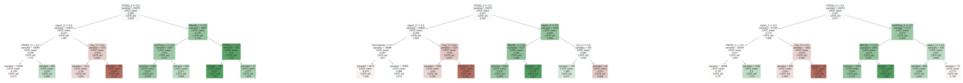


Figure 31: Original-Edu A Figure 32: WGAN-Edu A Figure 33: VAE-Edu A

Figure 34: Heterogeneous treatment effect charts from original, WGAN-50 and VAE-50 populations from EU-SILC Norway on treatment Education A from Causal Forest. Each box in the diagram refers to a split in the tree. The honest approach indicates that one batch of data is used to design the splits while another batch is placed in the category. The red boxes indicate the group with a low CATE score that implies the negative effect of treatment (intervention), and the green boxes indicate the group with a high CATE score with a good response to treatment. The treatment (intervention) is education (PE040) split in a binary with less than three on variable PE040 as zero and above as one. The outcome is self-perceived health (PH010).

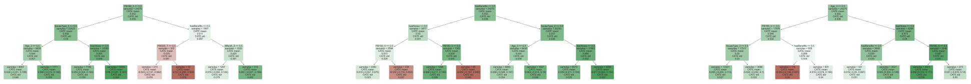


Figure 35: Original-Edu-B

Figure 36: WGAN-Edu-B

Figure 37: VAE-Edu B

Figure 38: Heterogeneous treatment effect charts from original, WGAN-50 and VAE-50 populations from EU-SILC Norway on treatment Education B from Causal Forest. Each box in the diagram refers to a split in the tree. The honest approach indicates that one batch of data is used to design the splits while another batch is placed in the category. The red boxes indicate the group with a low CATE score that implies the negative effect of treatment (intervention), and the green boxes indicate the group with a high CATE score with a good response to treatment. The treatment (intervention) is education (PE040) split in a binary with less than four on variable PE040 as zero and above as one. The outcome is self-perceived health (PH010).

5 Discussion

Some issues of relevance to high-attribute synthetic populations are omitted in this project for various reasons. Deep generative methods for creating populations were born out of challenges of sparse covariance matrices and the problem of scaling the computations when the number of attributes rises. The low dimensional latent representation in the deep generative method was believed to solve the so-called zero-cell problems. In this project, zero-cells are not discussed, being sampling zeros (individual records that could exist in a natural population but is absent) or structural zeros (being contradictory individual records that never could represent any person in a natural population) [17]. Neither are confidentiality issues addressed. These are all relevant to the applicability of synthetic populations derived from deep generative methods. The zero-cell problems have recently been described well by, i.e. Garrido [17]. Within the data access in this project, any further investigation in this direction would fail. The confidentiality challenges are a complicated field with its tradition of trades, and it would be unfair to make any claims regarding confidentiality within this project's scope. A third field that needs to be investigated is the perfection of machine learning models against the task of generating synthetic populations. The primary concern in this project is to look at how already applied deep generative methods can translate into public health, particularly public health education, through policy games. Therefore, no optimisation or search for alternative loss functions and their like was included in this project because such approaches are premature until the contextual understanding of what a synthetic population should be in policy analysis and planning in public health is clarified.

5.1 High-Attribute Population Synthesis

High attribute synthetic populations have a wide variety of current and future usage beyond the context of public health policy analysis and planning, be it in a game or a real-life application for exploring health outcomes from interventions. In this project, the focus has been on deep generative methods for creating synthetic populations. Deep generative techniques disrupt the evaluation regimes of synthetic populations in areas like microsimulation and agent-based modelling that can only handle some attributes. Standardised root means square errors or similar error metrics, and Pearson's correlation coefficient help understand how well a synthetic population adapts to the original data. R-squared measures to what extent the entire synthetic population and its single variables explain the original data. As pointed out, the fields generating synthetic populations for various purposes do not agree on methods or metrics to evaluate populations. This question is not getting more manageable when the number of attributes in a population increases. The experiments with self-supervised clustering using neural manifold clustering and embedding show different patterns in the original and replica populations. However, the technique must be more mature to expand from imaging to tabular data. However, if it does learn specific non-linear statistical patterns in the population, it can be a future tool for evaluating and better explaining synthetic populations. The Bland-Altman plots are excellent for a more informative evaluation of the synthetic populations than just RSMSE, Pearsons and R-squared. These plots show that all investigated deep generative models fail to keep all variables inside their confidence intervals, suggesting non-random variance that could impact analysis based on data from these populations. Like more traditional evaluation methods, the confusion and contingency matrices can help visualise differences and similarities between the original and synthetic populations. These methods can only be applied to one or two variables but help assess critical variables like the outcome variable self-perceived health in this project. Finding all poorly adapted correlations is like searching for the needle in the haystack. Given the identification of the underrepresentation of people with attributes poorly represented in the original, it is a great need to establish better methods for evaluating high-attribute synthetic populations. Synthetic populations meant for use in analysing interventions related to inequalities in health are in particular need of a method to explain better the disproportionate distribution of small (vulnerable) groups.

5.2 Scaling of Synthetic Populations

To power a policy game on public health, various methods to upscale population data to mimic the inhabitants of, i.e. a municipality would be of great interest. Adapting survey and census data to a geographical region is within the domain of microsimulation. Nevertheless, they are concerned with low-attribute data and use methods that do not scale to high attributes. Scaling by increasing the number of generated examples enhanced the identified misrepresentations, making this upscaling prone to losing the original population's heterogeneity. Scaling by first training models to produce regions on the EU-SILC for Norway only helped by keeping the regions more clearly defined. It solved the problem of underrepresenting Region Three but resulted in a total population with less similarity to the original than obtained by generating the population in one go from the original data. The shift in scales and truncation of variables, giving basement and ceiling effects, still destroy the options for keeping heterogeneity with any of these scaling methods. The challenges linked to these effects must be solved to get better representative populations. In a game setting, it is possible to apply more dirty tricks to achieve a population that, for educational purposes, would deal with inequality in health. However, the authenticity, understood as epidemiological strength, is lost without genuinely solving the challenges of, i.e. truncation and heterogeneity.

5.3 Causal Forest and Heterogeneous Treatment Effects

Comparing outcomes from heterogeneous treatment effects by Causal Forest DML between original and synthetic populations serve two purposes. First, a good match indicates a high authenticity to the educational goal. Second, a good match is required, but there are other requirements to reach a reasonable level of epidemiological standards.

The VAE-50, VAE-10 and WGAN-50 and WGAN-100 score well in a qualitative comparison between the charts of conditional average treatment effect from a Causal Forest DML model trained on original data. WGAN models do better than VAE models measured by the ranking in Shap plots. These evaluations are qualitative, as no original to one synthetic data record is available. The goal of the evaluation has been to look for the best matches by picking the correct variables for split and correctly getting the correct labelling as red or green boxes in the chart. VAE and WGAN with 50 or 100 in latent representation dimension did well, but no model matched perfectly. The matching is good news for the educational use of the synthetic population by the deep generative method. Causal Forest DML showcases a policy intervention on, i.e. inequality in health, by allowing for analysis based on observational data to predict which groups would favour or not (improve self-reported health) from a particular intervention. However, the reduced heterogeneity discussed above still needs to improve to secure educational goals of policy analysis and planning for inequality in health.

In order to reach an epidemiological standard, synthetic population generation techniques must deal with all the challenges mentioned above. In addition, the explainability must improve to understand the differences between the original data and the replica inherent to a specific generative method. Of course, creating a synthetic population that copies all statistical structures of the original while keeping heterogeneity is perhaps impossible. However, future evaluations of synthetic populations should be embedded in their use context. Applying Causal Forest DML has been one way to embed the use context of policy analysis and planning to reduce inequalities in health and provide a framework for evaluation in the same go. While the initial plan for the project was to create synthetic data from the Norwegian HUNT data bank and re-run previous epidemiological analyses to evaluate epidemiological similarity, the opportunity to explore Causal Forest DML as a substitute has been highly fruitful. Reproducing epidemiological studies with synthetic data is powerful, however, limited in scope to the context of the re-run study. Causal Forest DML, on the other hand, is a data-driven tool that both lends itself to epidemiology and not. In this project, the ”and not” implies that no matter the original data's epidemiological quality, synthetic populations generated with these original data can be benchmarked against a tool like Causal Forest. The choice of covariates (all other variables than the outcome and treatment variables) is highly relevant in epidemiology but only matters if the purpose is to compare an original dataset with a synthetic.

5.4 Synthetic Populations for Policy Games

According to epidemiological strength, educational authenticity to a synthetic population for use in a policy game has been discussed. A policy game may or may not be educational. The suggested deep generative model produces populations based on actual data that can bring realism to a game. Some entertaining games, like the Fishing series from the Norwegian company Misc Games, apply governmental economic and environmental policies behind the scenes in their gameplay. Synthetic populations have the potential to give life to non-player characters in a game. EU-SILC data for most EU countries that are freely available are de-anonymised with extremely small outliers removed. These data can be used as is in a game, with no generation of a synthetic counterpart. However, interesting individual data can be available in the future when methods can guarantee confidentiality. For this to happen, the questions of heterogeneity and its like and anonymisation must be addressed in research.

Agent-based modelling has been used in games. Agent-based modelling has synthetic populations at its core to feed its agents with attributes. The lessons learned in this project can be helpful to these communities if the need for high-attributed agents arises.

6 Conclusion

Deep generative methods can produce good high-attribute synthetic populations from individual records with health, social and welfare data like EU-SILC. The scores on standardised root mean squared error, Pearson's correlation coefficient and R-squares are good. An excellent tool for visualising the quality of a high-attribute synthetic population is the Blant Altman, suggested by researchers in microsimulation and agent-based modelling like [48] but only sometimes used.

Synthetic populations meant for analysis, simulations or games in public health requiring health outputs require an excellent reconstruction of the heterogeneity in the original data. Unfortunately, the variational autoencoder and Wasserstein adversarial network with gradient penalty produce some variables with low reliability due to non-random errors in variance, as shown in the Bland Altman plots. Furthermore, truncating data and shifts in scale metrics compared to original data threaten the applicability of these populations within public health and are mainly related to policy analysis and planning dealing with inequalities in health. The result is that heterogeneity in the original data is partly lost, and examples from small groups in the original data get underrepresented in the synthetic populations. Scaling of the populations by increasing the number of generated examples or training on regions data to be merged increases the discrimination following from the deep generative methods' lack of reproducing the heterogeneity in the original data.

In a policy game on public health for combatting inequalities in health, the lack of heterogeneity can be mitigated by the creative use of deep generative methods to do deliberate heterogeneity sampling. However, then, the epidemiological quality of the data needs to be recovered. In order to provide sufficiently heterogeneous high-attribute synthetic populations from actual individual data, the deep generative methods should be better explained, and the problems of truncating in such ways as leading to the underrepresentation of small (vulnerable) groups should be solved. Until then, the evaluation of synthetic populations for public health policy analysis and planning in or outside a game should be broadened to cover the potential impact on health outcomes and the discriminating effects of using the data. Such evaluation can guide countermeasures while waiting for the research communities to bring better high-attribute synthetic populations.

References

[1] Martin Arjovsky and Léon Bottou. “Towards Principled Methods for Training Generative Adversarial Networks”. eng. In: *arXiv* (2017).

[2] Susan Athey and Stefan Wager. “Estimating treatment effects with causal forests: An application”. In: *Observational Studies* 5.2 (2019), pp. 37–51.

[3] Godwin Badu-Marfo, Bilal Farooq, and Zachary Patterson. “Composite Travel Generative Adversarial Networks for Tabular and Sequential Population Synthesis”. In: *IEEE transactions on intelligent transportation systems* (2022), pp. 1–10. ISSN: 1524-9050. DOI: 10.1109/TITS.2022.3168232.

[4] Dimitris Ballas, Tom Broomhead, and Phil Mike Jones. “Spatial microsimulation and agent-based modelling”. In: *The Practice of Spatial Analysis* (2019), pp. 69–84.

[5] R Bonita, R Beaglehole, and T Kjellström. “Basic Epidemiology”. In: *Basic epidemiology. 2nd ed ed. Geneva: World Health Organization* (2006). URL: https://apps.who.int/iris/handle/10665/43541.

[6] Stanislav S. Borysov, Jeppe Rich, and Francisco C. Pereira. “How to generate micro-agents? A deep generative modeling approach to population synthesis”. In: *Transportation research. Part C, Emerging technologies* 106 (2019), pp. 73–97. ISSN: 0968-090X. DOI: 10.1016/j.trc.2019.07.006.

[7] Victor Chernozhukov et al. *Generic Machine Learning Inference on Heterogenous Treatment Effects in Randomized Experiments*. 2017. DOI: 10.48550/ARXIV.1712.04802. URL: https://arxiv.org/abs/1712.04802.

[8] Yu-Liang Chou et al. “Counterfactuals and causability in explainable artificial intelligence: Theory, algorithms, and applications”. In: *Information Fusion* 81 (2022), pp. 59–83.

[9] TD Cook and DT Campbell. “Quasi-experimentation: design and analysis issues for field settings”. In: (1979).

[10] Göran Dahlgren and Margaret Whitehead. *European strategies for tackling social inequities in health: Levelling up Part 2*. 2006. URL: https://apps.who.int/iris/handle/10665/107791.

[11] Göran Dahlgren and Margaret Whitehead. “Levelling up (part 1): a discussion paper on concepts and principles for tackling social inequities in health”. In: (2006). URL: https://apps.who.int/iris/handle/10665/107790.

[12] Göran Dahlgren and Margaret Whitehead. “The Dahlgren-Whitehead model of health determinants: 30 years on and still chasing rainbows”. eng. In: *Public health (London)* 199 (2021), pp. 20–24. ISSN: 0033-3506.

[13] Evans T Diderichsen F Whitehead M. “The Social Basis of Disparities in Health”. eng. In: *Challenging Inequities in Health*. New York: Oxford University Press, 2001. ISBN: 9780195137408.

[14] Carl Doersch. *Tutorial on Variational Autoencoders*. 2016. DOI: 10.48550/ARXIV.1606.05908. URL: https://arxiv.org/abs/1606.05908.

[15] ec.europa.eu/eurostat. *Eurostat EU-SILC*. 2023. URL: https://ec.europa.eu/eurostat/web/microdata/european-union-statistics-on-income-and-living-conditions (visited on 02/21/2023).

[16] econml.azurewebsites.net. *Econml*. 2023. URL: https://econml.azurewebsites.net/ (visited on 04/11/2023).

[17] Sergio Garrido et al. "Prediction of rare feature combinations in population synthesis: Application of deep generative modelling". In: *Transportation Research Part C: Emerging Technologies* 120 (2020).

[18] Ian J Goodfellow et al. "Generative Adversarial Networks". eng. In: *arXiv* (2014).

[19] Bart N. Green, Claire D. Johnson, and Alan Adams. "Writing narrative literature reviews for peer-reviewed journals: secrets of the trade". eng. In: *Journal of chiropractic medicine* 5.3 (2006), pp. 101–117. ISSN: 1556-3707.

[20] Helse- og omsorgsdepartementet. *Lov om folkehelsearbeid (folkehelseloven)*. https://lovdata.no/dokument/NL/lov/2011-06-24-29. Accessed: 2022-11-07. 2011.

[21] Mohammad S Jalali et al. "Evolution and Reproducibility of Simulation Modeling in Epidemiology and Health Policy Over Half a Century". eng. In: *Epidemiologic reviews* 43.1 (2022), pp. 166–175. ISSN: 1478-6729.

[22] Chapuis Kevin, Taillandier Patrick, and Drogoul Alexis. "Generation of Synthetic Populations in Social Simulations: A Review of Methods and Practices". In: *Journal of artificial societies and social simulation* 25.2 (2022). ISSN: 1460-7425. DOI: 10.18564/jasss.4762.

[23] Diederik P Kingma and Max Welling. "Auto-Encoding Variational Bayes". eng. In: *arXiv* (2014).

[24] Zengyi Li et al. "Neural manifold clustering and embedding". In: *arXiv preprint arXiv:2201.10000* (2022).

[25] Ellicott C. Matthay and M. Maria Glymour. "Causal Inference Challenges and New Directions for Epidemiologic Research on the Health Effects of Social Policies". eng. In: *Current epidemiology reports* 9.1 (2022), pp. 22–37. ISSN: 2196-2995.

[26] I. S. Mayer. "The gaming of policy and the politics of gaming: a review". In: *Simulation & gaming* 40.6 (2009), pp. 825–862. ISSN: 1046-8781. DOI: 10.1177/1046878109346456.

[27] Igor Mayer. "Gaming for policy analysis. Learning about complex multi-actor systems". In: *Why do games work* (2008), pp. 31–40.

[28] World Health Organization. *Essential Public Health Functions, Health Systems, and Health Security - Developing Conceptual clarity and a WHO roadmap for action.* 2018. URL: https://apps.who.int/iris/bitstream/handle/10665/272597/9789241514088-eng.pdf (visited on 05/29/2023).

[29] Judea Pearl. "Radical empiricism and machine learning research". eng. In: *Journal of causal inference* 9.1 (2021), pp. 78–82. ISSN: 2193-3677.

[30] Azizur Rahman. "Estimating small area health-related characteristics of populations: a methodological review". In: *Geospatial Health* 12.1 (2017).

[31] Wojciech Roszka. "Spatial microsimulation of personal income in Poland at the level of subregions". In: *Statistics in Transition new series* 20.3 (2019), pp. 133–153.

[32] Brent D Ruben. "Simulations, games, and experience-based learning: The quest for a new paradigm for teaching and learning". In: *Simulation & Gaming* 30.4 (1999), pp. 498–505.

[33] Christian Elling Scheele, Ingvild Little, and Finn Diderichsen. "Governing health equity in Scandinavian municipalities: The inter-sectorial challenge". eng. In: *Scandinavian journal of public health* 46.1 (2018), pp. 57–67. ISSN: 1403-4948.

[34] sikt.no. *Sikt Forskningsdata*. 2023. URL: https://sikt.no/omrade/forskningsdata (visited on 01/10/2023).

[35] Eric Silverman et al. "Situating agent-based modelling in population health research". In: *Emerging Themes in Epidemiology* 18.1 (2021), pp. 1–15.

[36] Dianna M Smith, Alison Heppenstall, and Monique Campbell. "Estimating Health over Space and Time: A Review of Spatial Microsimulation Applied to Public Health". eng. In: *J* 4.2 (2021), pp. 182–192. ISSN: 2571-8800.

[37] Niko Speybroeck et al. "Simulation models for socioeconomic inequalities in health: A systematic review". eng. In: *International journal of environmental research and public health* 10.11 (2013), pp. 5750–5780. ISSN: 1661-7827.

[38] H.P.E.M Spitters et al. "Developing a policy game intervention to enhance collaboration in public health policymaking in three European countries". eng. In: *BMC public health* 17.1 (2017), pp. 961–961. ISSN: 1471-2458.

[39] Timo Szczepanska et al. "GAM on! Six ways to explore social complexity by combining games and agent-based models". In: *International Journal of Social Research Methodology* (2022), pp. 1–15.

[40] Keith S Taber. "Mediated learning leading development—The social development theory of Lev Vygotsky". In: *Science education in theory and practice: An introductory guide to learning theory* (2020), pp. 277–291.

[41] Robert Tanton et al. "A review of spatial microsimulation methods". In: *International Journal of Microsimulation* 7.1 (2014), pp. 4–25.

[42] Aleid Sunniva Teeuwen et al. "A systematic review of the impact of food security governance measures as simulated in modelling studies". In: *Nature Food* (2022).

[43] A. Whitworth et al. “Estimating uncertainty in spatial microsimulation approaches to small area estimation: A new approach to solving an old problem”. eng. In: *Computers, environment and urban systems* 63 (2017), pp. 50–57. ISSN: 0198-9715.

[44] Phil Wilkinson. “A brief history of serious games”. In: *Entertainment Computing and Serious Games: International GI-Dagstuhl Seminar 15283, Dagstuhl Castle, Germany, July 5-10, 2015, Revised Selected Papers*. Springer. 2016, pp. 17–41.

[45] Megan R Winkler et al. “Applications of Complex Systems Models to Improve Retail Food Environments for Population Health: A Scoping Review”. In: *Advances in Nutrition* 13.4 (2022), pp. 1028–1043.

[46] Jiao Xi and James P Lantolf. “Scaffolding and the zone of proximal development: A problematic relationship”. In: *Journal for the Theory of Social Behaviour* 51.1 (2021), pp. 25–48.

[47] Yu Xiao and Maria Watson. “Guidance on Conducting a Systematic Literature Review”. eng. In: *Journal of planning education and research* 39.1 (2019), pp. 93–112. ISSN: 0739-456X.

[48] Boyam Fabrice Yaméogo et al. “Generating a two-layered synthetic population for French municipalities: Results and evaluation of four synthetic reconstruction methods”. In: *JASSS-Journal of Artificial Societies and Social Simulation* 24.2 (2021). ISSN: 1460-7425. DOI: 10.18564/jasss.4482.

[49] Yaodong Yu et al. “Learning diverse and discriminative representations via the principle of maximal coding rate reduction”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 9422–9434.

Appendix

A Glossary

- *AAPD* Adjusted Average Predicted Difference is a frequency metric for how many variables fall in or out of a bounded area (usually the confidence interval around original variables) around the truth value (original data).
- *Latent dimension* Latent dimension or latent representation is a more condensed form of the data, with a shape less than the original. Any layer in a neural network smaller than the original can represent a compressed input form. The size of this compressed input representation is called the latent dimension.
- *Pearson's* correlation coefficient is a measure of the match between two variables.
- *Policy* ”A set of ideas or a plan of what to do in particular situations that have been agreed to officially by a group of people, a business organisation, a government, or a political party.” (Britannica): ”An officially accepted set of rules or ideas about what should be done”. (Cambridge Dictionary)
- *Public Health* ”Public health is an organised effort by society, primarily through its public institutions, to improve, promote, protect and restore the population's health through collective action.” (WHO)
- *Public health policy* is defined as the laws, regulations, plans, decisions and actions implemented within society in order to promote wellness and ensure that specific health goals are met. (Derived from WHO and used as definition in this project 2.2)
- *R-squared* measures how much the model explains a feature. In this case, how much of the synthetic population attributes explain the original data attributes?
- *RAE* Relatively Absolute Error is the TAE divided by the number of attributes. These measures depend on the variables' scales. Useful for comparing the differences in marginals between models.
- *Reliability* 1) the quality of being trusted or believed because of working or behaving well. 2) how well a machine, equipment, or system works. 3) how accurate or able to be trusted someone or something is considered to be. (Cambridge Dictionary)
- *SRMSE* Standard Root Mean Squared Error.
- *Sampling zeros* Original data misses examples that are present in the entire population. The so-called sampling zeros problem is partly solved if the synthetic population captures some

of these examples. Refered to in Garrido et al. [17].

- *Standard deviation* Gaussian standard deviation.
- *Structural zeros* Original data never has examples absent in the entire population. Synthetic population by deep generative methods may contain impossible examples in the entire population. Refered to in Garrido et al. [17].
- *TAE* Total Absolute Error is the absolute difference between original (true) and synthetic features (predicted). This measure varies according to the variables scale and number of examples, and hence not useful for other than measuring differences between models, i.e., VAE and WGAN.
- *Validity* 1) the quality of being based on truth or reason or being accepted. 2) the state of being acceptable or reasonable. (Cambridge Dictionary)
- *Z-scores* are the Gaussian normalising of a variable setting mean to 0. It is calculated by subtracting the mean from the value and dividing it by the standard deviation for a variable.

```
TAE = abs(predicted - true)
RAE = 1/n-variables * TAE
AAPD = 1/n-variables * abs(counts inside - counts outside)
Standard deviation = sqrt(1/N \Sigma (true - predicted) ** 2)
Pearson's correlation coefficient = cov(true, predict) /
(standard deviation(true) * standard deviation (predicted))
Z-score = (value - mean(all values)) / standard deviation
```

B Code

A Variational Autoencoders

A variational autoencoder [23] is an autoencoder with an intermediate probabilistic component between the encoder and decoder. While a simple autoencoder has two, often mirrored, neural networks directly connected, the variational autoencoder has an intermediate component that learns Gaussian parameters before passing its output to the decoder. When run through the decoder, these parameters are applied to a randomly sampled vector that will produce an output replica similar to the original input to the encoder. The intermediate component executes the kernel trick performed by passing z_mean, which is the latent representation as a dense layer, and z_log_var is an additional dense layer, both receiving the same input (see code below). The loss function is a combination of the Kullback-Leibler divergence $KL(q_\phi(z|x), p(z) = -\Sigma_{j=1}^{J}(1 + log(\sigma^2) - \mu^2 - \sigma^2$ at the intermediate component and binary cross entropy $H_p(q) = -\frac{1}{N}\Sigma_{i=1}^{N} y_i * log(p(y_i)) + (1 - y_i) * log(1 - p(y_i))$ for the final output from the decoder. The Kullback-Leibler divergence forces the latent variable to maintain a Gaussian distribution, while the binary cross entropy captures the reconstruction loss between the original and replica.

The kernel trick in variational autoencoders creates a probability prediction for each cell in the latent layer instead of the more straightforward 0 or 1 representation in a regular autoencoder. This makes the variational autoencoders perform significantly better than autoencoders in reproducing their inputs.

A.1 Code for VAE

The code for the variational autoencoder used in this project. After training, the encoder and decoder can be used separately to feed replicas of an original data record to train the Neural Manifold Clustering and Embedding algorithm.

```
import torch
from torch import nn


class VAE(nn.Module):
    def __init__(self,
                 feature_dimension,
                 latent_dimension):
        super(VAE, self).__init__()
        self.encoder = nn.Sequential(
            nn.Linear(feature_dimension, 100),
            self._block(100, 150),
            nn.Linear(150, latent_dimension)
        )
        self.z_mean = nn.Linear(latent_dimension, latent_dimension)
        self.z_log_var = nn.Linear(latent_dimension, latent_dimension)
        self.decoder = nn.Sequential(
            nn.Linear(latent_dimension, 150),
            self._block(150, 100),
            nn.Linear(100, feature_dimension),
            nn.Sigmoid()
        )
    def _block(self, input_d, n_nodes):
        return nn.Sequential(
            nn.Linear(in_features=input_d, out_features=n_nodes),
            nn.BatchNorm1d(n_nodes),
            nn.LeakyReLU(0.2))

    def encode(self, x):
        x = self.encoder(x)
        mu = self.z_mean(x)
        log_var = self.z_log_var(x)
        return mu, log_var

    def get_latent(self, x):
        latent = self.encoder(x)
        return latent

    def decode(self, z):
        return self.decoder(z)

    def forward(self, x):
        mu, log_var = self.encode(x)
        epsilon = torch.randn_like(log_var)
        z_parametrised = epsilon * (torch.exp(log_var / 2)) + mu
        x = self.decode(z_parametrised)
        return [x, mu, log_var]
```

The code for training in a notebook:

```
import numpy as np
import pandas as pd
import torch
import torch.nn as nnfrom torch.utils.data import DataLoader
import torch.optim as optim
# Custom class to clean and reshape data and create and
#  reshape synthetic data from the model output.
from src.data_cleaning import DataClean as dc
```

```
torch.manual_seed(42)
number_epochs = 80
learning_rate = 1e-4
optimiser = optim.RMSprop(model_vae.parameters(), lr=learning_rate)
loss_fn = nn.BCELoss(reduction="sum")
batch_size = 128
latent_dimension = 50
feature_dimension = df.shape[1]
beta_vae = 0.5

data = dc(data_file, prepared, config_file)
df = data.get_data()  # all one-hots

def kl_loss(mu, log_var):
    loss = - 0.5 * torch.sum(1 + log_var -
                             torch.exp(log_var) -
                             mu ** 2)
    return loss

model_vae = VAE(df.shape[1], latent_dimension)
model_vae.train()
torch_data = torch.tensor(df.values, dtype=torch.float32)
loader = DataLoader(torch_data, batch_size=batch_size, shuffle=True)
for epoch in range(number_epochs):
    for batch_idx, (real) in enumerate(loader):
        replica, z_mean, z_sigma = model_vae(real)
        reconstruction_loss = loss_fn(replica, real)
        kl = beta_vae * (kl_loss(z_mean, z_sigma) / real.shape[1])
        loss = reconstruction_loss + kl
        optimiser.zero_grad()
        loss.backward()
        optimiser.step()
        if batch_idx % 50 == 0 and batch_idx > 0:
            print(f"Epoch [{epoch} / {number_epochs}] \ "
                  f"KL Loss: {kl:4f}, Rep Loss: {reconstruction_loss:.4f}")
```

```
# When the model is trained:
df_rc = data.get_data_recategorised()
torch_s = model_vae.decoder(torch.randn(df.shape[0], latent_dimension)
# Synthetic data as a pandas data frame (all one-hots)
df_synthetic = data.get_synthetic(torch_s.detach().numpy(), columns=df_rc.columns)
```

B Generative Adversarial Networks

B.1 Code for WGAN-GP

Code for the Wasserstein generative adversarial network with gradient penalty used in the project.

```
"""
WGAN-GP
Generative adversarial networks for synthetic population generation
using Wasserstein and gradient penalty to stabilise.
"""
import torch
import torch.nn as nn



class Critic(nn.Module):
    def __init__(self, feature_dimension, output_dim=1):
        super(Critic, self).__init__()
        self.feature_dimension = feature_dimension
        self.critic = nn.Sequential(
            self._block(self.feature_dimension, 100),
            self._block(100, 150),
            nn.Linear(in_features=150, out_features=output_dim),
        )

    def _block(self, input_d, n_nodes):
        return nn.Sequential(
            nn.Linear(in_features=input_d, out_features=n_nodes),
            # do not use batch-norm in critic
            nn.InstanceNorm1d(n_nodes),
            nn.LeakyReLU(0.2))
    def forward(self, x):
        return self.critic(x)


class Generator(nn.Module):
    def __init__(self, feature_dimension, latent_dimension):
        super(Generator, self).__init__()
        self.latent_dimension = latent_dimension
        self.feature_dimension = feature_dimension
        self.generator = nn.Sequential(self._block(self.latent_dimension, 150),
                                       self._block(150, 100),
                                       nn.Linear(100, self.feature_dimension),
                                       nn.Sigmoid())

    def forward(self, x):
        return self.generator(x)

    def _block(self, input_d, n_nodes):
        return nn.Sequential(
            nn.Linear(in_features=input_d, out_features=n_nodes),
            # nn.LayerNorm(n_nodes),
            nn.BatchNorm1d(n_nodes),
            nn.LeakyReLU(0.2))


def initialise_weights(model):
    for m in model.modules():
        if isinstance(m, nn.Linear):
            nn.init.normal_(m.weight.data, 0.0, 0.02)


def gradient_penalty(model, real, fake):
    batch_size = real.shape[0]
    feature_dimension = real.shape[1]
    # One epsilon per example
    epsilon = torch.rand(batch_size, 1).repeat(1, feature_dimension)
    interpolated = (real * epsilon + fake * (1 - epsilon))
    mixed_score = model(interpolated)
    gradient = torch.autograd.grad(inputs=interpolated,
                                   outputs=mixed_score,
                                   grad_outputs=torch.ones_like(mixed_score),
                                   create_graph=True,
                                   retain_graph=True)[0]
    gradient = gradient.view(gradient.shape[0], -1)  # flatten
    gradient_norm = torch.linalg.vector_norm(gradient, ord=2, dim=1)
    gp = torch.mean((gradient_norm - 1) ** 2)
    return gp
```

Code for training the model in a notebook.

```
import numpy as np
import pandas as pd
import torch
from torch.utils.data import DataLoader
import src.data_cleaning import DataClean as dc

torch.manual_seed(42)
number_epochs = 250
learning_rate = 1e-4
batch_size = 128
latent_dimension = 50
feature_dimension = df.shape[1]
critic_iterations = 5
lambda_gp = 10.0
beta_1 = 0.5
beta_2 = 0.9
critic = Critic(feature_dimension, output_dim=1)
generator = Generator(feature_dimension, latent_dimension)
initialise_weights(critic)
initialise_weights(generator)
opt_critic = optim.Adam(critic.parameters(), lr=learning_rate, betas=(beta_1, beta_2))
```

```
opt_generator = optim.Adam(generator.parameters(), lr=learning_rate, betas=(beta_1, beta_2))

generator.train()
critic.train()

data = dc(data_file, prepared, config_file) # Custom data handling class
df = data.get_data()
torch_data = torch.tensor(df.values, dtype=torch.float32)
loader = DataLoader(torch_data, batch_size=batch_size, shuffle=True)
for epoch in range(number_epochs):
    for batch_idx, (real) in enumerate(loader):
        # Train critic (max log(critic(real)) + (1 - log(critic(z)))
        for _ in range(critic_iterations):
            noise = torch.randn((real.shape[0], latent_dimension))
            fake = generator(noise)
            critic_real = critic(real)
            critic_fake = critic(fake)
            gp = gradient_penalty(critic, real, fake)
            # Set minus in front of optimising equation --> to maximise
            loss_critic = - (torch.mean(critic_real) - torch.mean(critic_fake))
            loss_critic += lambda_gp * gp
            critic.zero_grad()
            loss_critic.backward()
            opt_critic.step()
        # Train generator (min log(1 -  critic(gen(z))) max log(critic(gen(z)))
        # min --> - E[critic(generator(fake))]
        fake = generator(noise)
        logits_fake = critic(fake)
        loss_generator = - torch.mean(logits_fake)
        generator.zero_grad()
        loss_generator.backward()
        opt_generator.step()
        if epoch % 10 == 0:
            collect_loss.append((loss_critic, loss_generator, gp))
        if batch_idx % 50 == 0 and batch_idx > 0:
            print(f"Epoch [{epoch} / {number_epochs}] \ "
                    f"Loss C: {loss_critic:4f}, Loss G: {loss_generator:.4f}, GP: {gp}")
```

C Neural Manifold Clustering and Embedding

Manifold learning is to map data points to a low-dimensional representation that preserves the manifold structure in the data [24]. The idea of manifold clustering and embedding is that if data points come from a union of low-dimensional manifolds, it is possible to segment the data points based on their corresponding manifolds to obtain a low-dimensional embedding for each manifold [24]. Principal component analysis can extract manifolds of linear subspaces, one of the most basic forms of unsupervised learning (Jolliffe 1986 in [24]). More challenging clustering problems concern unions of non-linear low-dimensional manifolds. Neural manifold clustering and embedding (NMCE) are proposed as a solution for neural networks to solve these clustering problems [24]. The technique combines data record augmentation and the algorithm called maximal coding rate reduction (Yu et al. 2020 in [24]). Unsupervised learning of categories of objects in images performed better than the current state-of-the-art methods for subspace clustering [24].

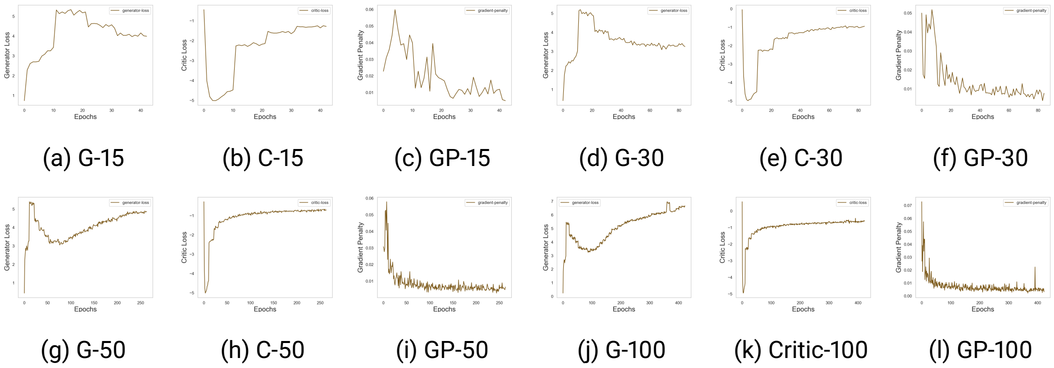
Next, the question is if this method of unsupervised learning of categories can properly categorise individual data records relevant to public health.

C.1 Code for NMCE

Code used to run the Neural Manifold Clustering and Embedding are used out of the box as described in the original article [24]. The model is run with PyTorch in a notebook like this:

```
n_steps = 2000
print_every = 300
bs = 1929
# One chunk is original data the other has synthetic data
# Can perhaps mix more synthetic
n_chunks = 2     # One for original and one for synthetic
amb_dim = 230    # Input dimension = number of variables
lat_dim = 150    # Neurons at each layer
z_dim = 100      # Latent layer
n_clusters = 20  # Number of extracted classes
lambda_ = 40     # Do not influence much
# Set up the NMCE model (code in source reference)
net = MLP_net(amb_dim, lat_dim, z_dim, n_clusters)
optimiser = optim.Adam(net.parameters(), lr=0.001, betas=(0.9,0.99), weight_decay=0.00001)
```

```
# Using NMCE's original softmax function
G_Softmax = Gumble_Softmax(0.2, straight_through=False)
# Using NMCE's implementation of MCR
criterion = MaximalCodingRateReduction(eps=0.01, gamma=1.0)
# Using NMCE's implementation of similarity
criterion_z = Z_loss()
begin = time.time()
for i in range(n_steps):
    # Use DataLoader to create batches
    loader = iter(DataLoader(dataset=x_data, batch_size=bs, shuffle=True))
    # Run one batch and update grads
    for j in range(len(loader)):
        x = next(loader)
        # Create augmented data from vae-model
        aug_latent = encoder(x.numpy())
        xn = decoder(aug_latent)
        xn = torch.tensor(np.array(xn), dtype=torch.float32)
        xt = torch.cat((xn, x), dim=0).float()
        z, logits = net(xt)
        loss_z, z_sim = criterion_z(z)
        z_sim = z_sim.mean()
        prob = G_Softmax(logits)
        z, prob = chunk_avg(z, n_chunks=n_chunks,
                            normalize=True), chunk_avg(prob, n_chunks=n_chunks)
        loss, loss_list= criterion(z,prob,num_classes=n_clusters)
        loss += lambda_ * loss_z
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()
    if i % print_every == 0:
        print('{} steps done, loss c {}, loss d {}, z sim {}'.format(i+1,loss_list[0],
              loss_list[1],z_sim.item()))
duration = time.time() - begin
print(duration)
```



(a) G-15 (b) C-15 (c) GP-15 (d) G-30 (e) C-30 (f) GP-30

(g) G-50 (h) C-50 (i) GP-50 (j) G-100 (k) Critic-100 (l) GP-100

Figure 39: Metrics for WGAN models used on EU-SILC Norway. G-loss is the training loss from the generator. C-loss is the training loss from the critic and GP-loss is the loss from gradient penalty. The last number is the size of the latent dimension for the model. Similar shaped loss curves are measured for EU-SILC Finland. The choice of models number of training iterations are taken from where the loss on generator and critic converge (flattens out).

C Deep Generative Models Convergence Plots

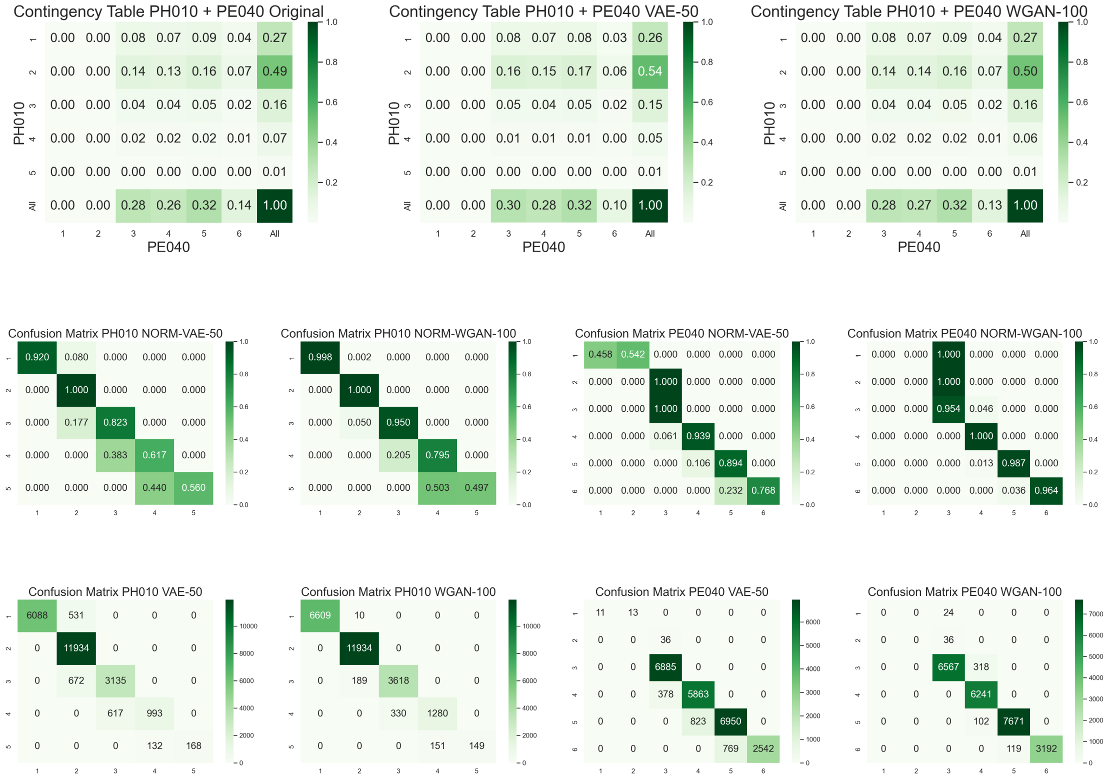
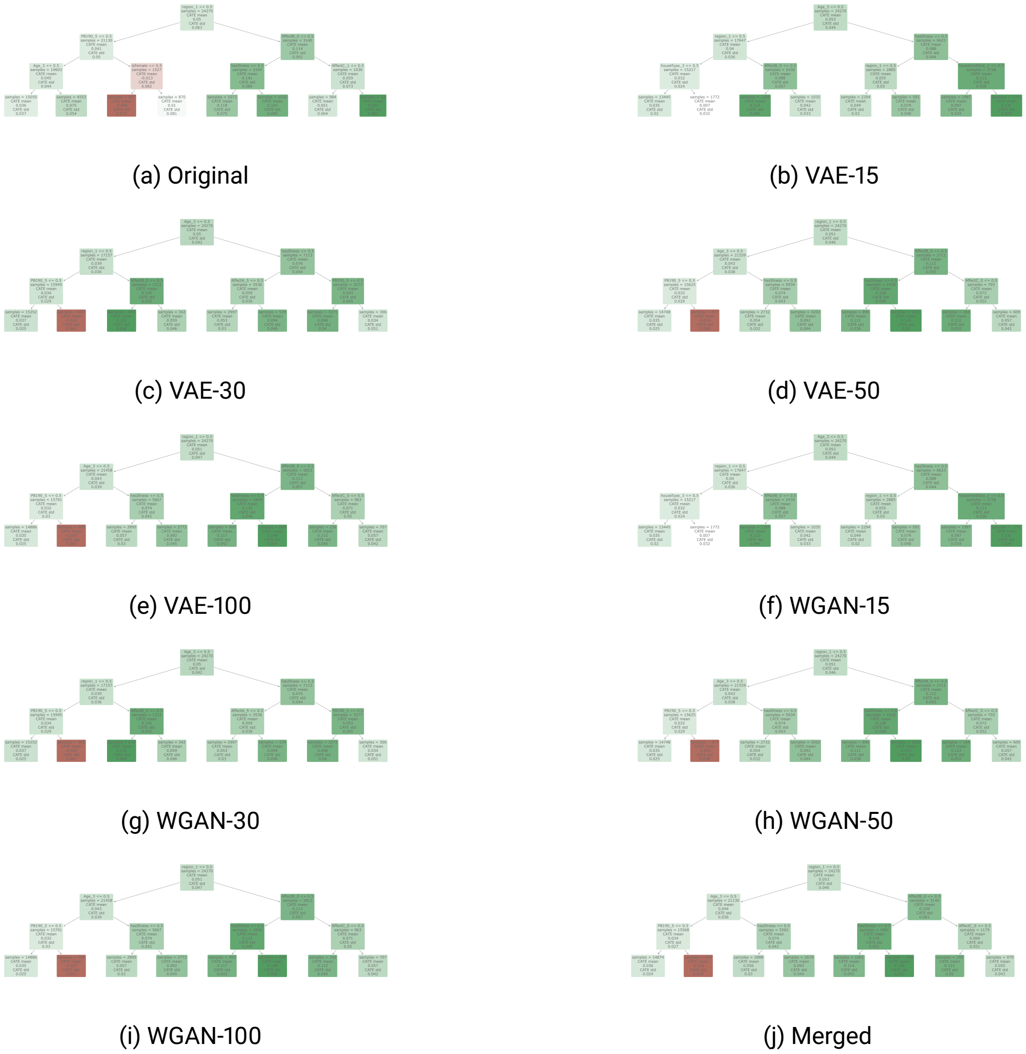


Figure 40: Confusion and contingency matrices for VAE-50 and WGAN-100. VAE-50 is chosen as the worst of the best models, and WGAN-100 as the best for comparison. Correlations between PH010 and PE040 are within a single population calculated by chi-square. For single variables PH010 and PE040, respectively, the synthetic populations are compared to the original data. Original data correct classification is represented by rows at the y-axis.

D Confusion and Contingency Matrices



(a) Original

(b) VAE-15

(c) VAE-30

(d) VAE-50

(e) VAE-100

(f) WGAN-15

(g) WGAN-30

(h) WGAN-50

(i) WGAN-100

(j) Merged

Figure 41: Decision charts in four levels from Causal Forets DML run on intervention ”education B” on data from EU-SILC Norway, showing all deep generative models and the synthetic population resulting from generating regions by VAE-100 separately and then merge them into a population.

(a) Original

(b) VAE-15

(c) VAE-30

(d) VAE-50

(e) VAE-100

(f) WGAN-15

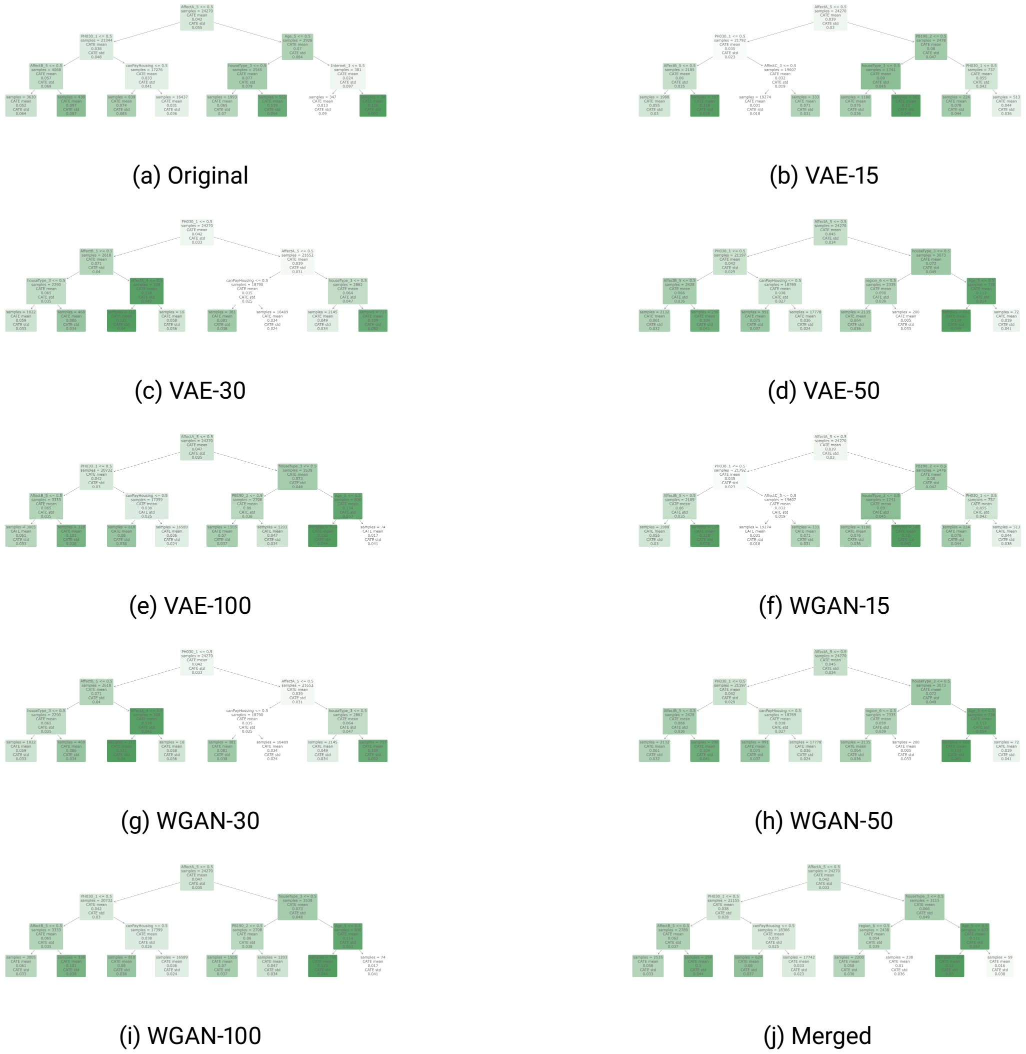
(g) WGAN-30

(h) WGAN-50

(i) WGAN-100

(j) Merged

Figure 42: Decision charts in four levels from Causal Forets DML run on intervention ”social” on data from EU-SILC Norway, showing all deep generative models and the synthetic population resulting from generating regions by VAE-100 separately and then merge them into a population.



(a) Original

(b) VAE-15

(c) VAE-30

(d) VAE-50

(e) VAE-100

(f) WGAN-15

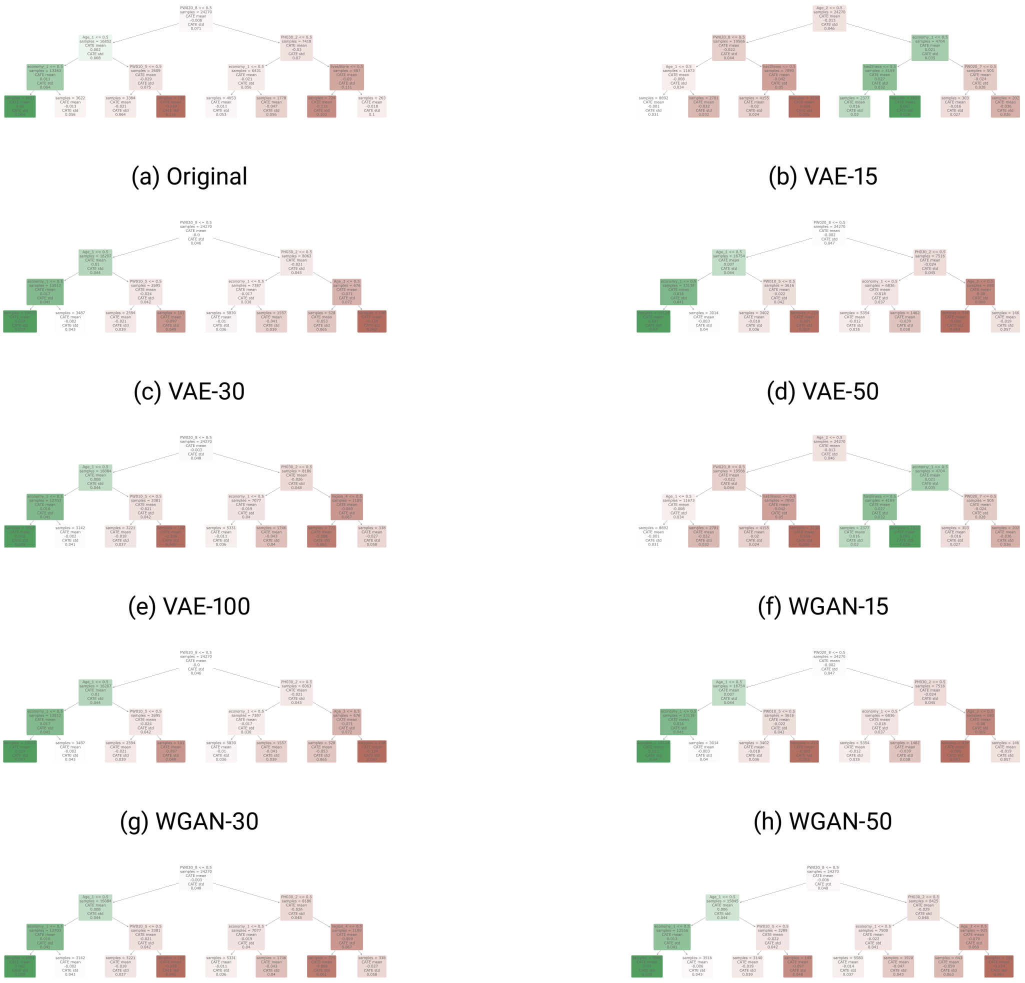
(g) WGAN-30

(h) WGAN-50

(i) WGAN-100

(j) Merged

Figure 43: Decision charts in four levels from Causal Forets DML run on intervention ”leisure” on data from EU-SILC Norway, showing all deep generative models and the synthetic population resulting from generating regions by VAE-100 separately and then merge them into a population.



(a) Original

(b) VAE-15

(c) VAE-30

(d) VAE-50

(e) VAE-100

(f) WGAN-15

(g) WGAN-30

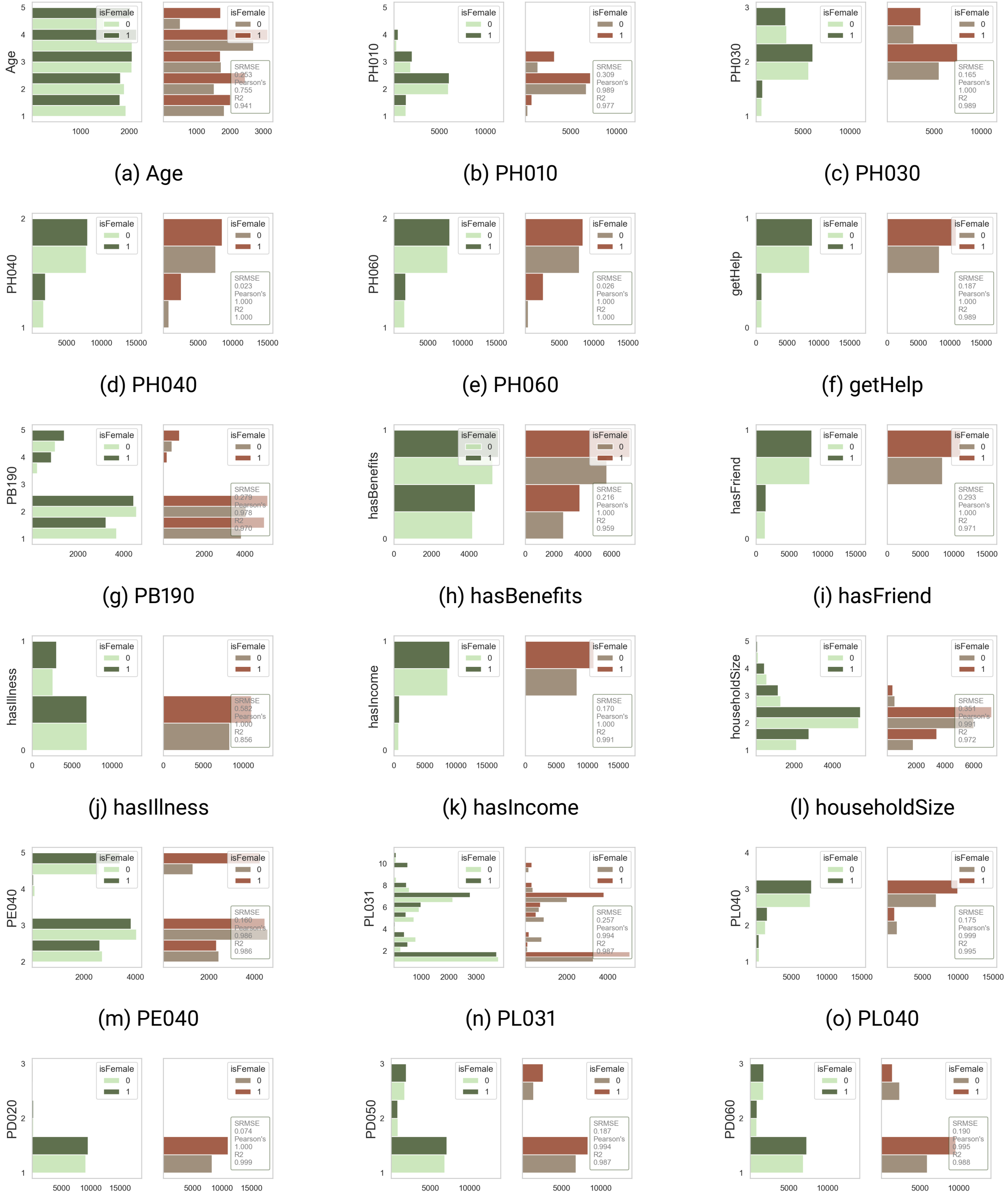
(h) WGAN-50

(i) WGAN-100

(j) Merged

Figure 44: Decision charts in four levels from Causal Forets DML run on intervention ”affect” on data from EU-SILC Norway, showing all deep generative models and the synthetic population resulting from generating regions by VAE-100 separately and then merge them into a population.

E Single Variables EU-SILC Finland 2013



(a) Age

(b) PH010

(c) PH030

(d) PH040

(e) PH060

(f) getHelp

(g) PB190

(h) hasBenefits

(i) hasFriend

(j) hasIllness

(k) hasIncome

(l) householdSize

(m) PE040

(n) PL031

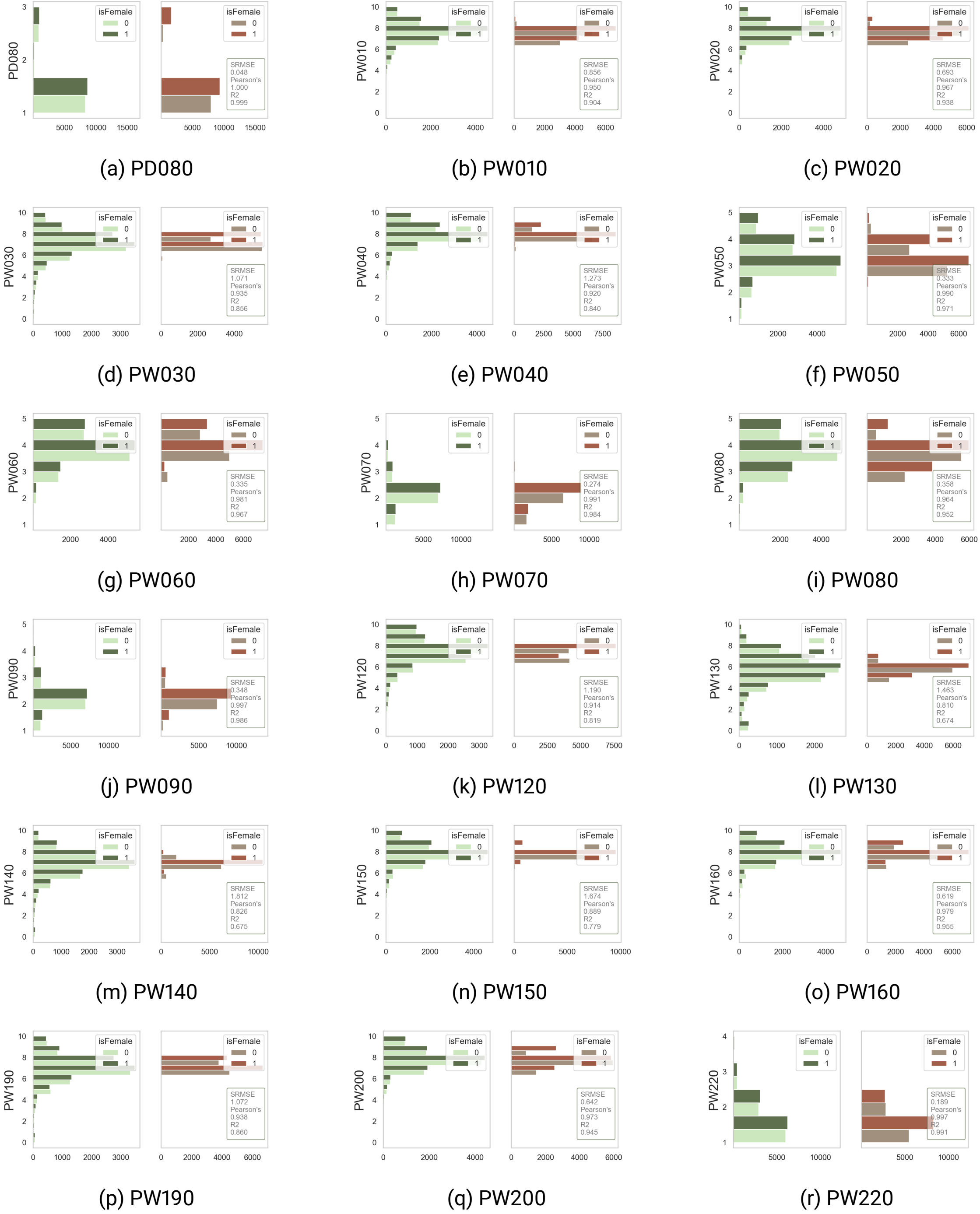
(o) PL040

(p) PD020

(q) PD050

(r) PD060

Figure 45: VAE-15 marginals from EU-SILC Finland illustrated with second variable gender. Green bars are from original and red bars from synthetic data. Differences on gender is visualised, and show the relationship between two variables. Metrics shown in figure is for difference between original and synthetic main variable.



(a) PD080 (b) PW010 (c) PW020

(d) PW030 (e) PW040 (f) PW050

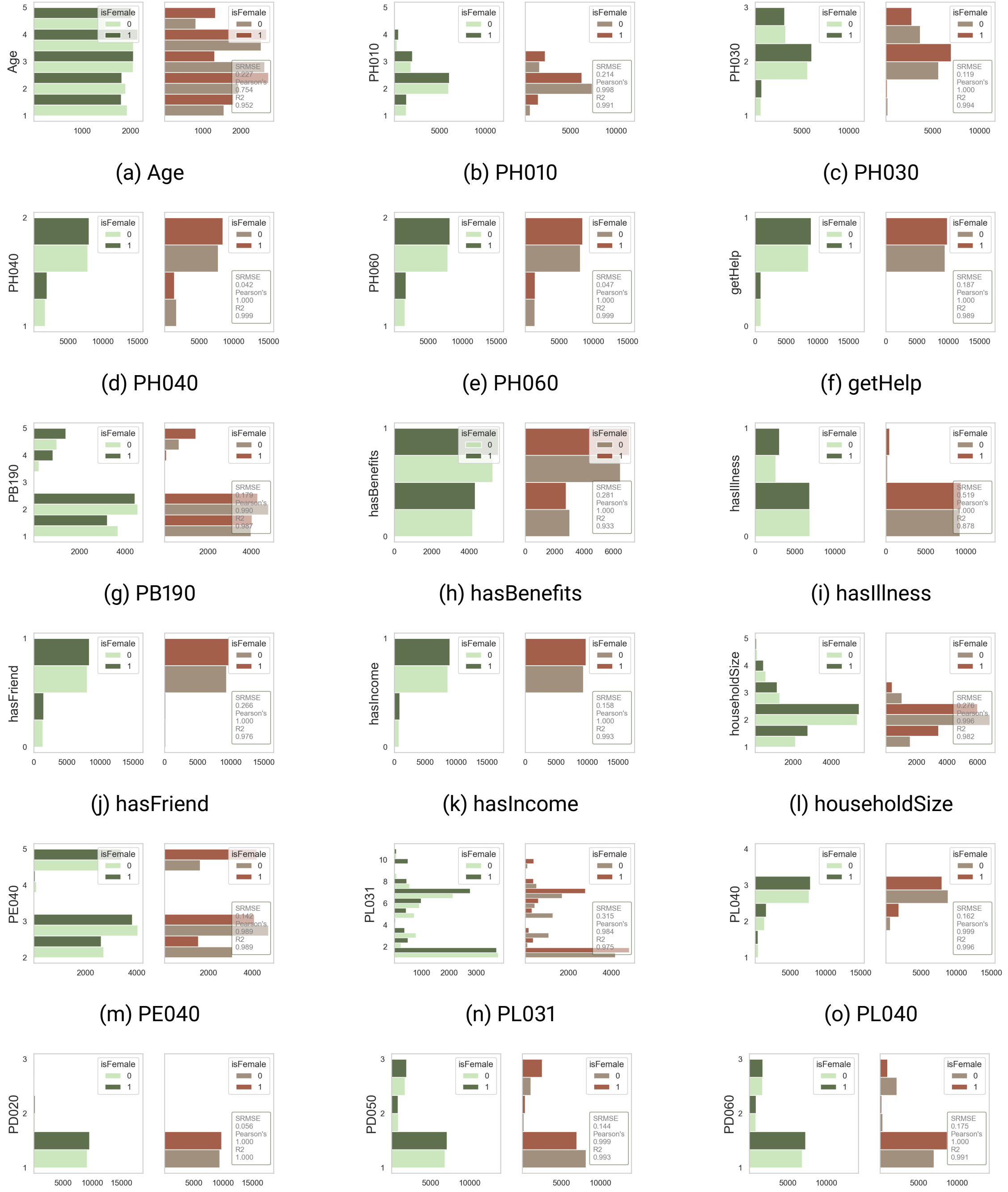
(g) PW060 (h) PW070 (i) PW080

(j) PW090 (k) PW120 (l) PW130

(m) PW140 (n) PW150 (o) PW160

(p) PW190 (q) PW200 (r) PW220

Figure 46: VAE-15 marginals from EU-SILC Finland illustrated with second variable gender. Green bars are from original and red bars from synthetic data. Differences on gender is visualised, and show the relationship between two variables. Metrics shown in figure is for difference between original and synthetic main variable.



(a) Age

(b) PH010

(c) PH030

(d) PH040

(e) PH060

(f) getHelp

(g) PB190

(h) hasBenefits

(i) hasIllness

(j) hasFriend

(k) hasIncome

(l) householdSize

(m) PE040

(n) PL031

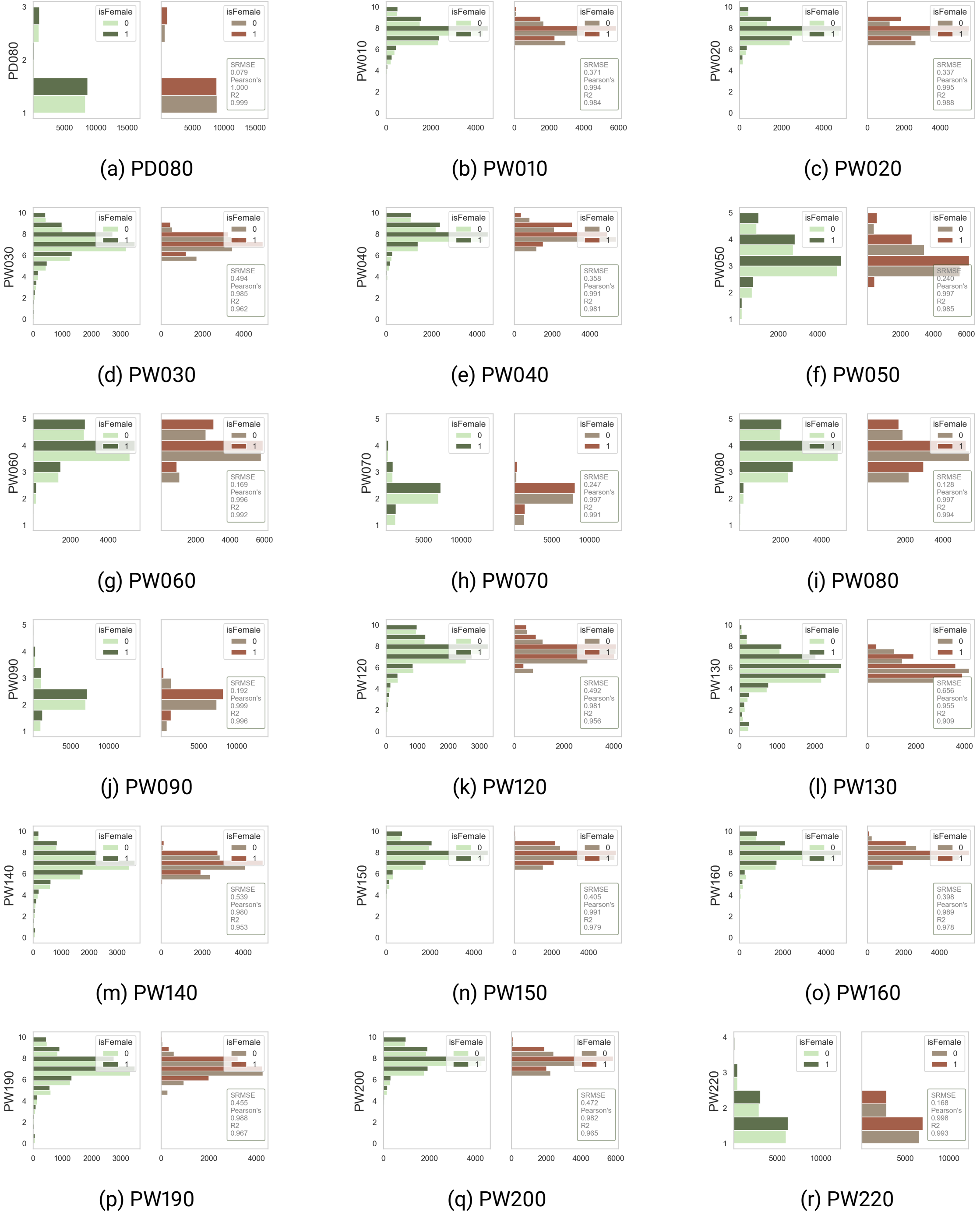
(o) PL040

(p) PD020

(q) PD050

(r) PD060

Figure 47: VAE-30 marginals from EU-SILC Finland illustrated with second variable gender. Green bars are from original and red bars from synthetic data. Differences on gender is visualised, and show the relationship between two variables. Metrics shown in figure is for difference between original and synthetic main variable.



(a) PD080

(b) PW010

(c) PW020

(d) PW030

(e) PW040

(f) PW050

(g) PW060

(h) PW070

(i) PW080

(j) PW090

(k) PW120

(l) PW130

(m) PW140

(n) PW150

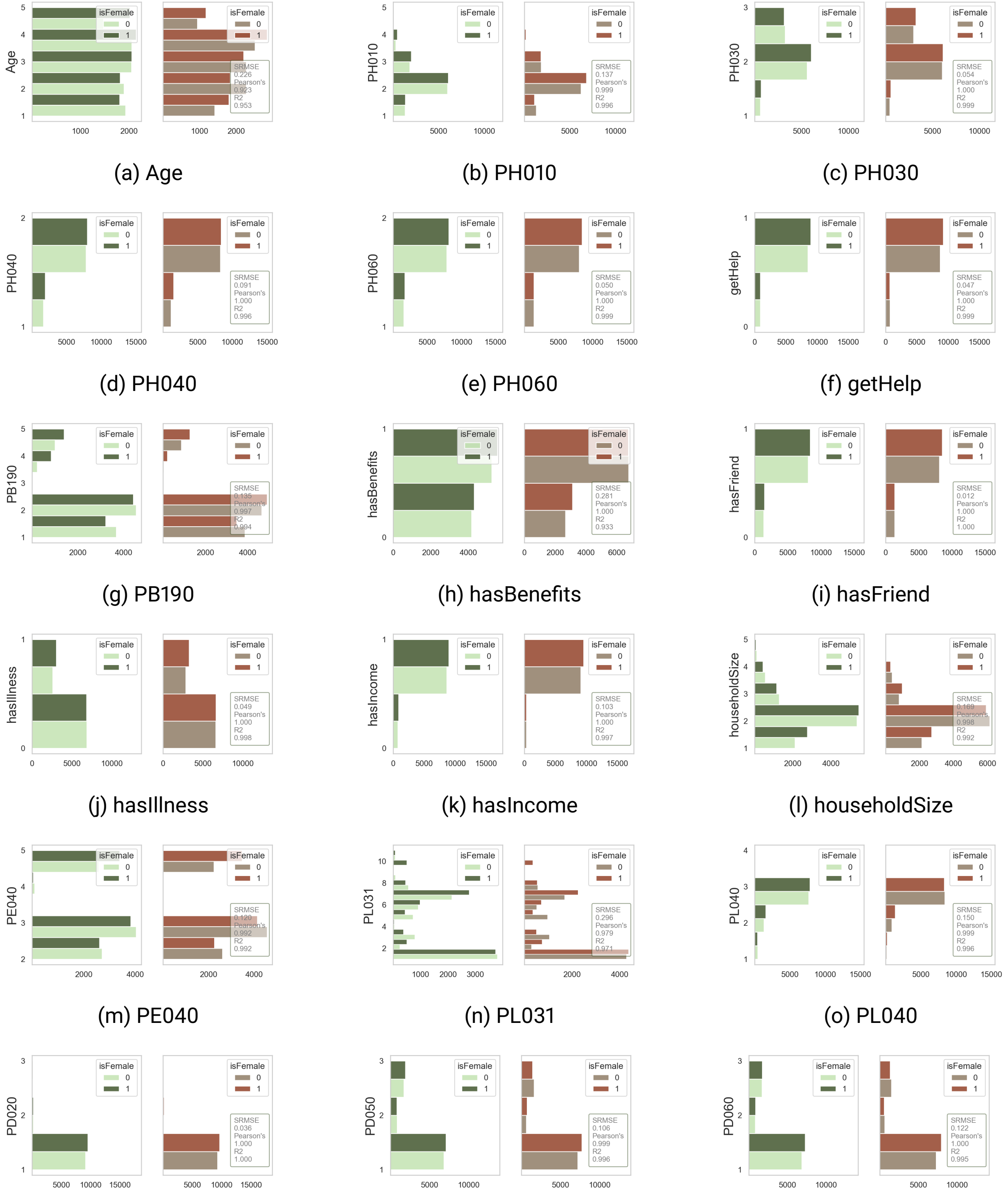
(o) PW160

(p) PW190

(q) PW200

(r) PW220

Figure 48: VAE-30 marginals from EU-SILC Finland illustrated with second variable gender. Green bars are from original and red bars from synthetic data. Differences on gender is visualised, and show the relationship between two variables. Metrics shown in figure is for difference between original and synthetic main variable.



(a) Age

(b) PH010

(c) PH030

(d) PH040

(e) PH060

(f) getHelp

(g) PB190

(h) hasBenefits

(i) hasFriend

(j) hasIllness

(k) hasIncome

(l) householdSize

(m) PE040

(n) PL031

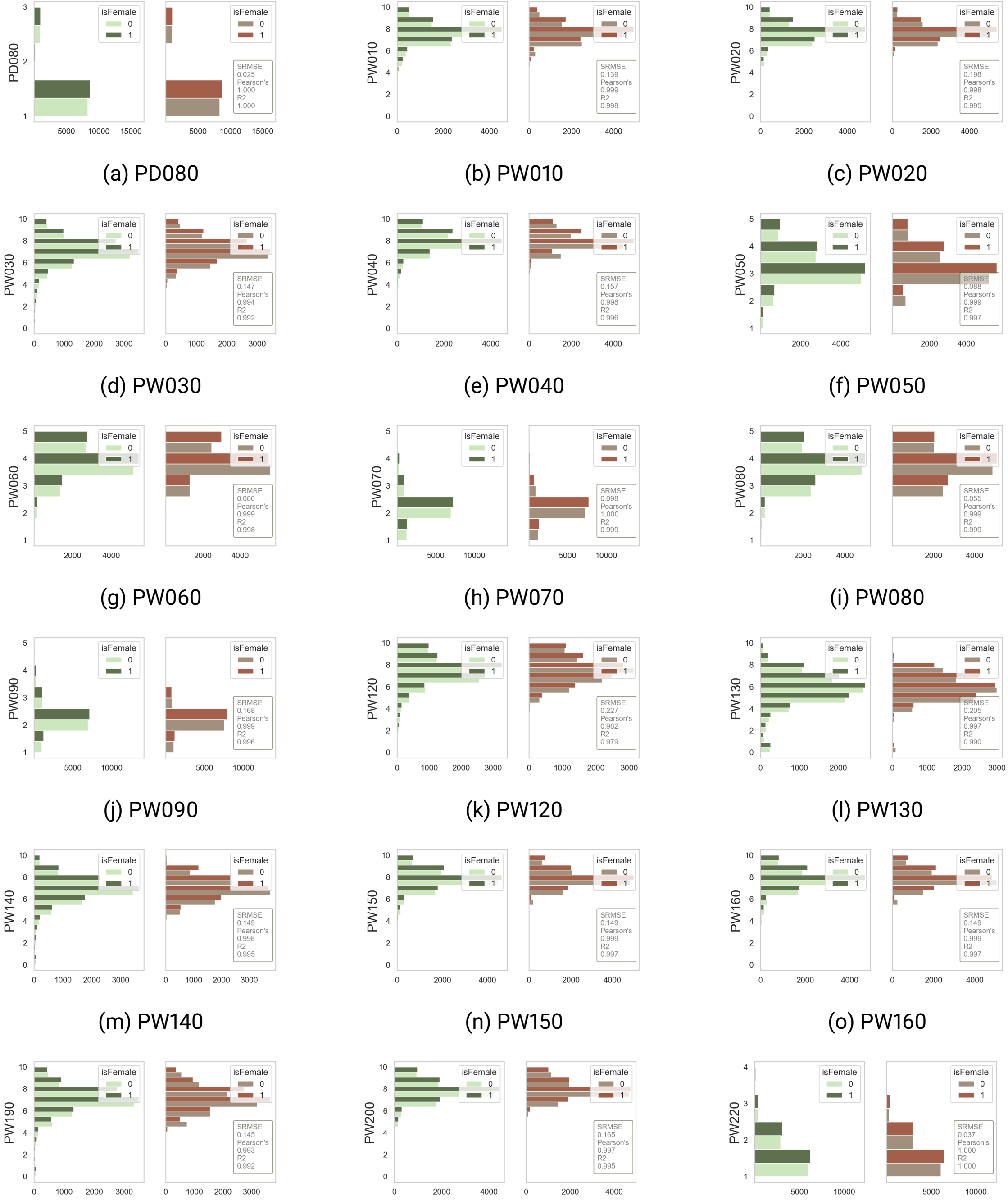
(o) PL040

(p) PD020

(q) PD050

(r) PD060

Figure 49: VAE-50 marginals from EU-SILC Finland illustrated with second variable gender. Green bars are from original and red bars from synthetic data. Differences on gender is visualised, and show the relationship between two variables. Metrics shown in figure is for difference between original and synthetic main variable.



(a) PD080

(b) PW010

(c) PW020

(d) PW030

(e) PW040

(f) PW050

(g) PW060

(h) PW070

(i) PW080

(j) PW090

(k) PW120

(l) PW130

(m) PW140

(n) PW150

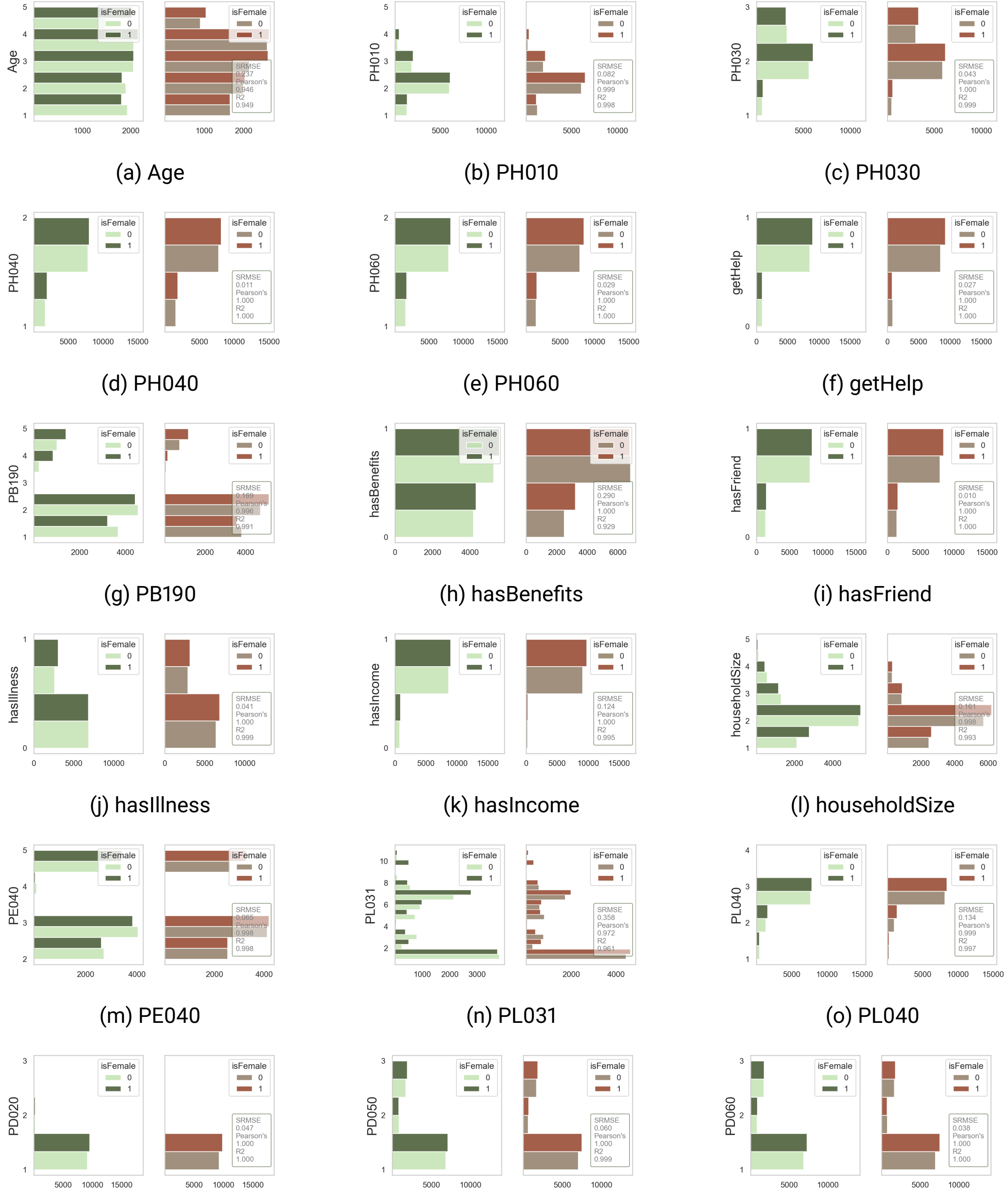
(o) PW160

(p) PW190

(q) PW200

(r) PW220

Figure 50: VAE-50 marginals from EU-SILC Finland illustrated with second variable gender. Green bars are from original and red bars from synthetic data. Differences on gender is visualised, and show the relationship between two variables. Metrics shown in figure is for difference between original and synthetic main variable.



(a) Age

(b) PH010

(c) PH030

(d) PH040

(e) PH060

(f) getHelp

(g) PB190

(h) hasBenefits

(i) hasFriend

(j) hasIllness

(k) hasIncome

(l) householdSize

(m) PE040

(n) PL031

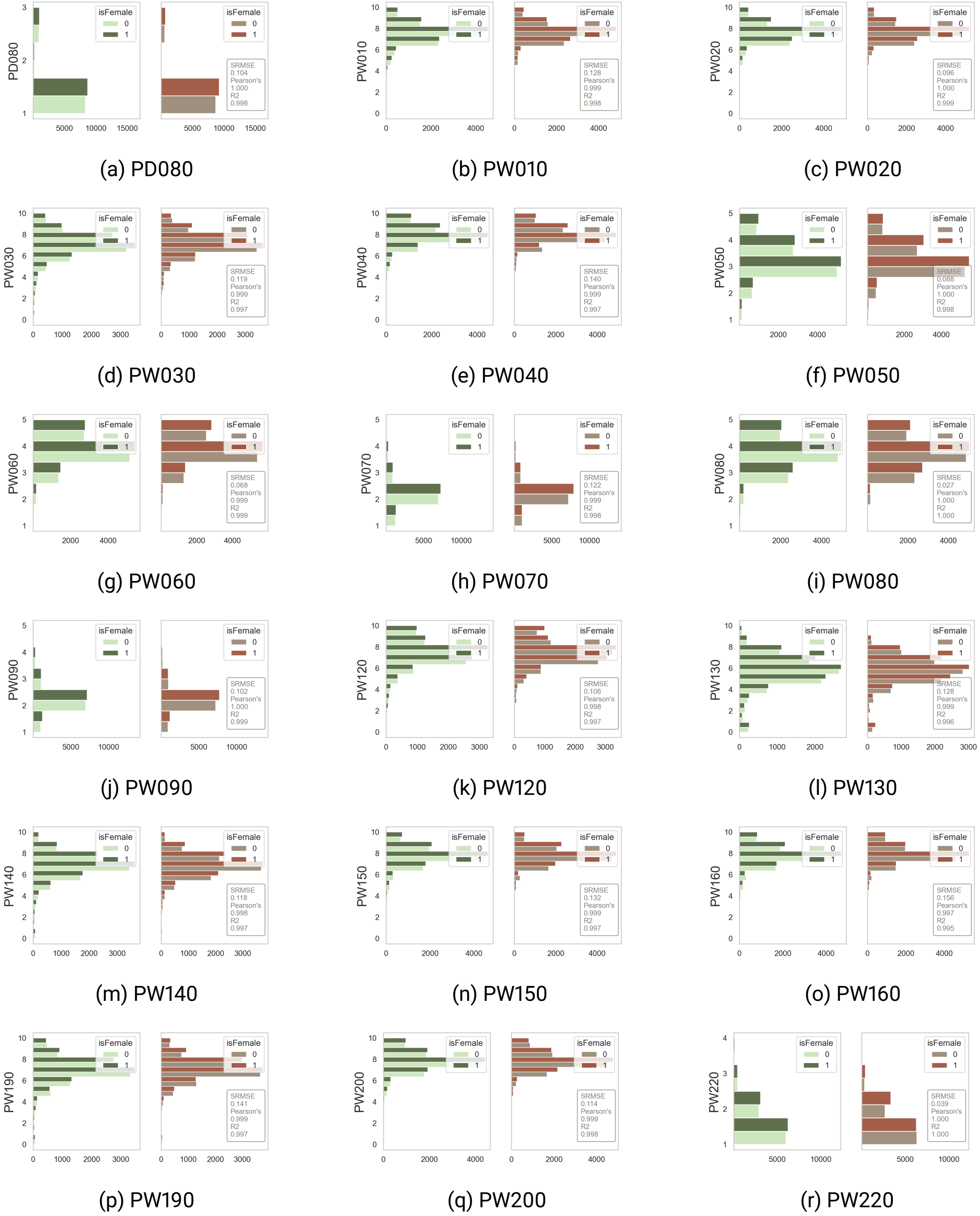
(o) PL040

(p) PD020

(q) PD050

(r) PD060

Figure 51: VAE-100 marginals from EU-SILC Finland illustrated with second variable gender. Green bars are from original and red bars from synthetic data. Differences on gender is visualised, and show the relationship between two variables. Metrics shown in figure is for difference between original and synthetic main variable.



(a) PD080

(b) PW010

(c) PW020

(d) PW030

(e) PW040

(f) PW050

(g) PW060

(h) PW070

(i) PW080

(j) PW090

(k) PW120

(l) PW130

(m) PW140

(n) PW150

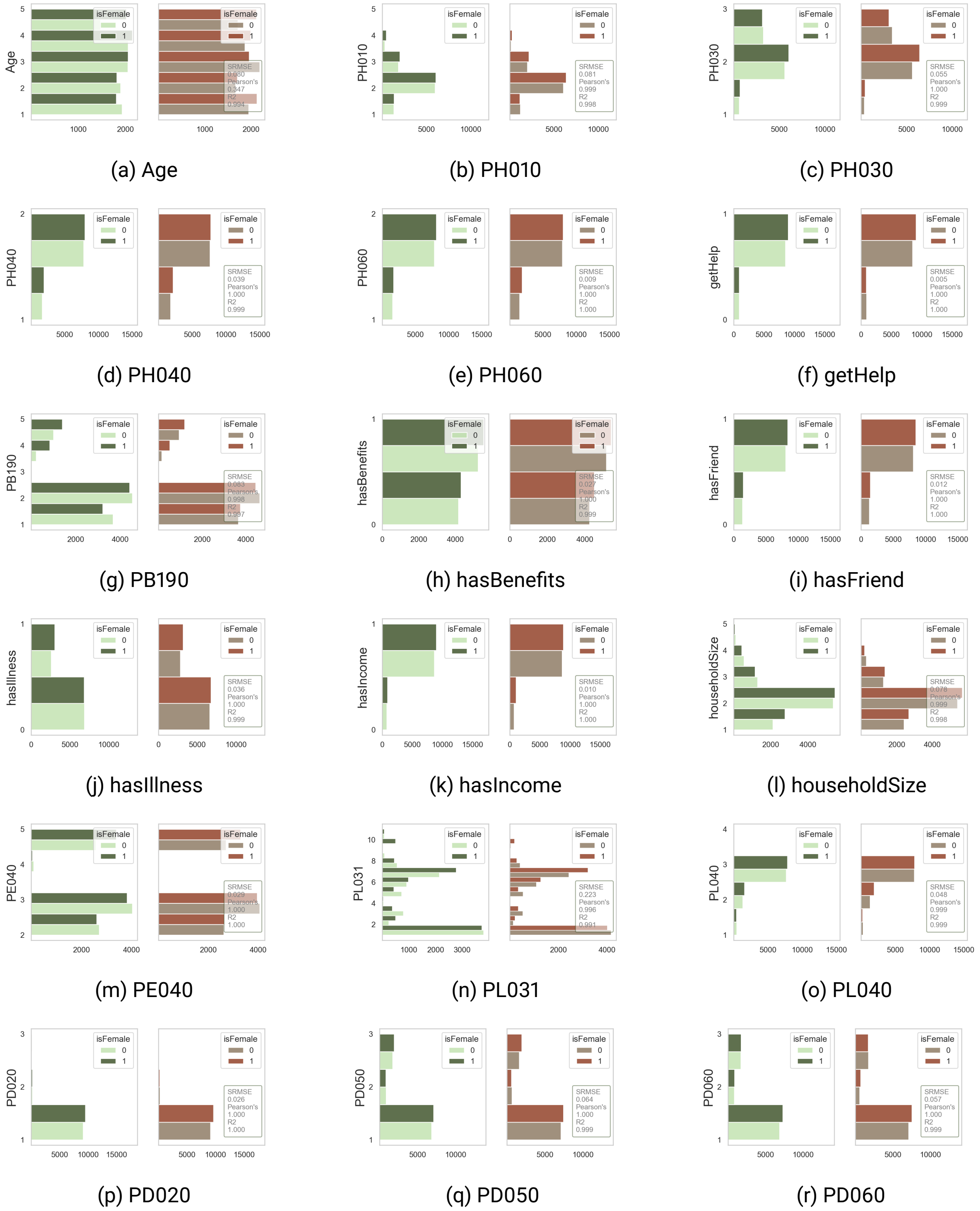
(o) PW160

(p) PW190

(q) PW200

(r) PW220

Figure 52: VAE-100 marginals from EU-SILC Finland illustrated with second variable gender. Green bars are from original and red bars from synthetic data. Differences on gender is visualised, and show the relationship between two variables. Metrics shown in figure is for difference between original and synthetic main variable.



(a) Age

(b) PH010

(c) PH030

(d) PH040

(e) PH060

(f) getHelp

(g) PB190

(h) hasBenefits

(i) hasFriend

(j) hasIllness

(k) hasIncome

(l) householdSize

(m) PE040

(n) PL031

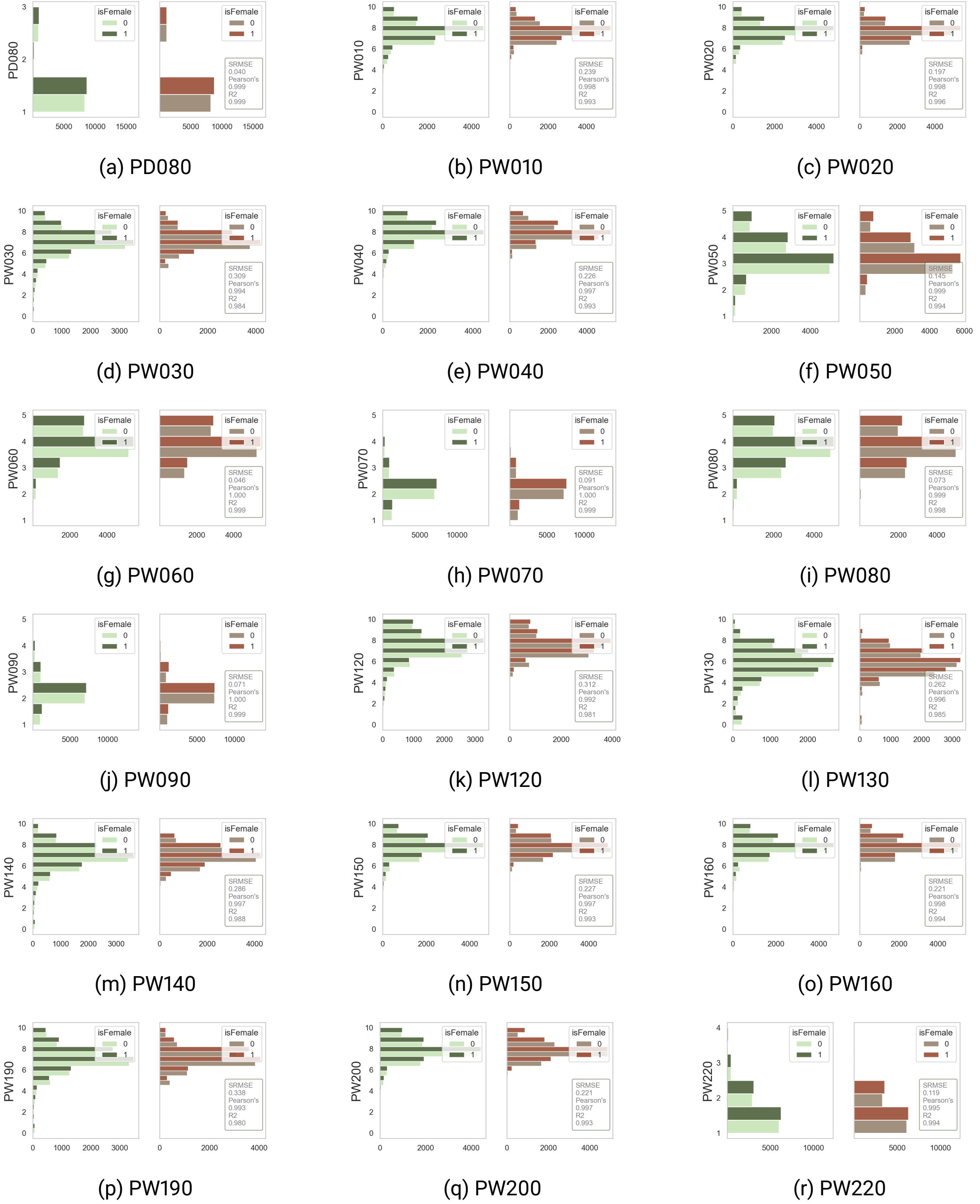
(o) PL040

(p) PD020

(q) PD050

(r) PD060

Figure 53: WGAN-15 marginals for all single variables from EU-SILC Finland visualised with second variable gender. Green bars are from original and red bars from synthetic data. Differences on gender is visualised, and show the relationship between two variables. Metrics shown in figure is for difference between original and synthetic main variable.



(a) PD080 (b) PW010 (c) PW020

(d) PW030 (e) PW040 (f) PW050

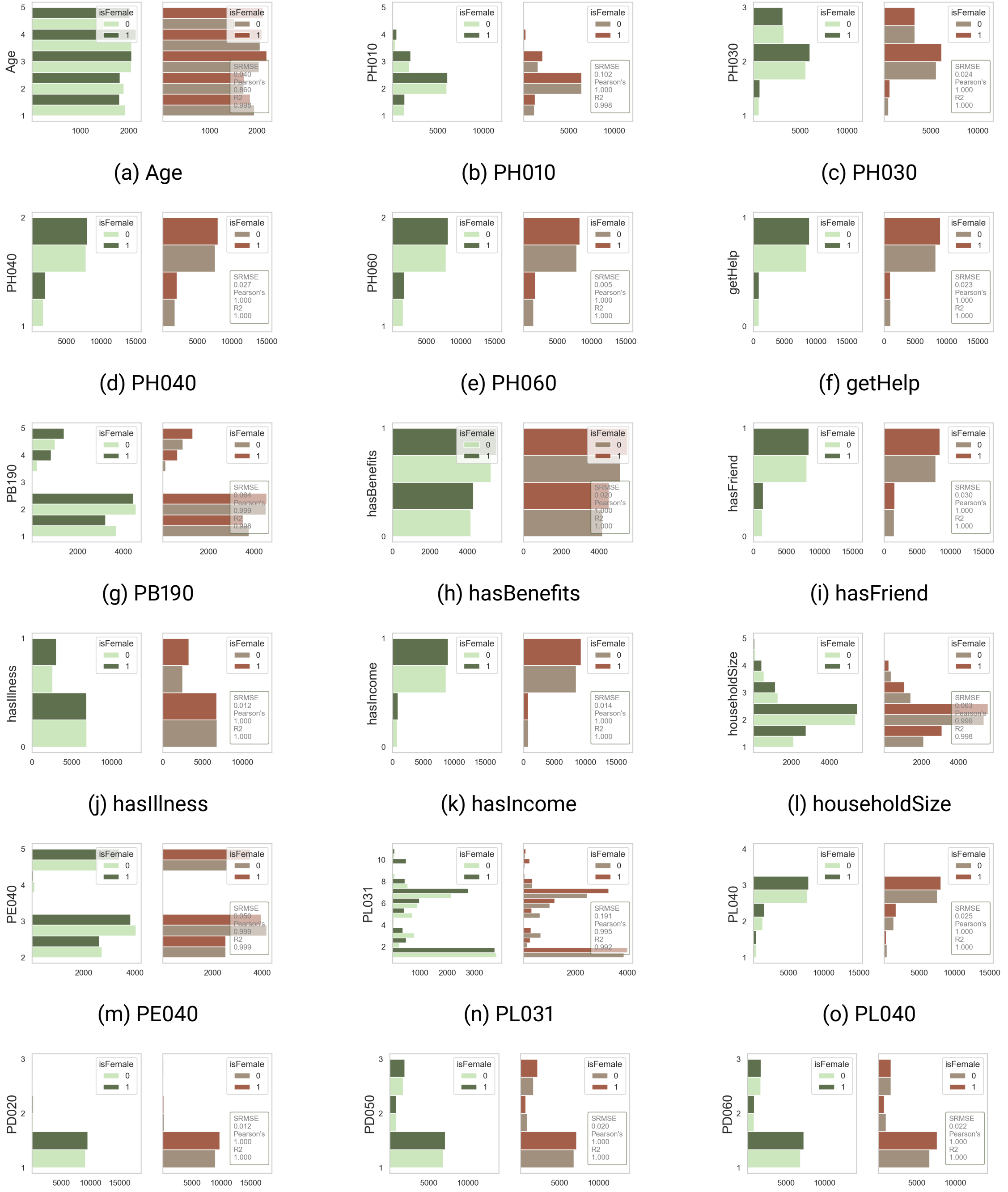
(g) PW060 (h) PW070 (i) PW080

(j) PW090 (k) PW120 (l) PW130

(m) PW140 (n) PW150 (o) PW160

(p) PW190 (q) PW200 (r) PW220

Figure 54: WGAN-15 marginals for all single variables from EU-SILC Finland visualised with second variable gender. Green bars are from original and red bars from synthetic data. Differences on gender is visualised, and show the relationship between two variables. Metrics shown in figure is for difference between original and synthetic main variable.



(a) Age

(b) PH010

(c) PH030

(d) PH040

(e) PH060

(f) getHelp

(g) PB190

(h) hasBenefits

(i) hasFriend

(j) hasIllness

(k) hasIncome

(l) householdSize

(m) PE040

(n) PL031

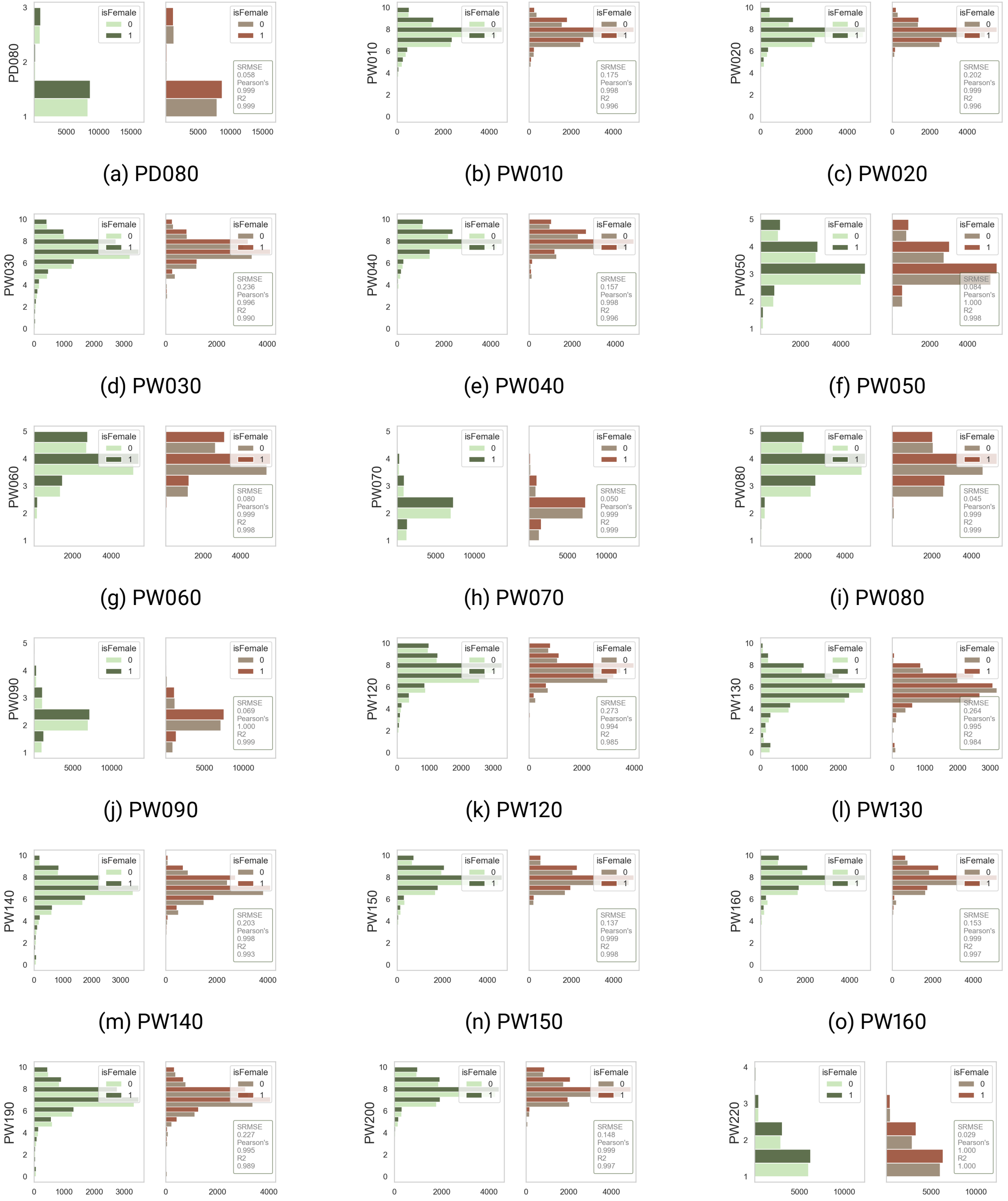
(o) PL040

(p) PD020

(q) PD050

(r) PD060

Figure 55: WGAN-30 marginals for all single variables from EU-SILC Finland visualised with second variable gender. Green bars are from original and red bars from synthetic data. Differences on gender is visualised, and show the relationship between two variables. Metrics shown in figure is for difference between original and synthetic main variable.



(a) PD080

(b) PW010

(c) PW020

(d) PW030

(e) PW040

(f) PW050

(g) PW060

(h) PW070

(i) PW080

(j) PW090

(k) PW120

(l) PW130

(m) PW140

(n) PW150

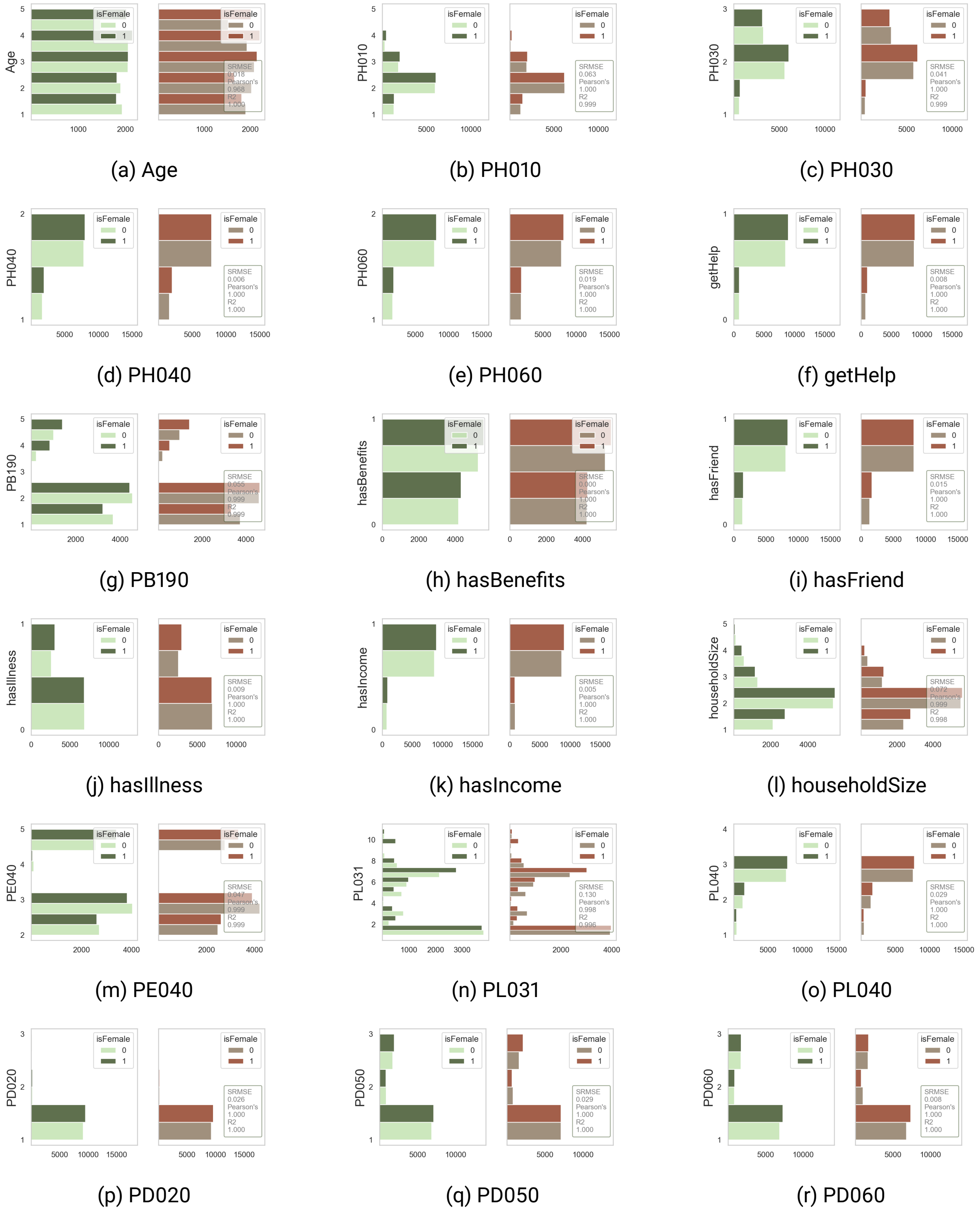
(o) PW160

(p) PW190

(q) PW200

(r) PW220

Figure 56: WGAN-30 marginals for all single variables from EU-SILC Finland visualised with second variable gender. Green bars are from original and red bars from synthetic data. Differences on gender is visualised, and show the relationship between two variables. Metrics shown in figure is for difference between original and synthetic main variable.



(a) Age

(b) PH010

(c) PH030

(d) PH040

(e) PH060

(f) getHelp

(g) PB190

(h) hasBenefits

(i) hasFriend

(j) hasIllness

(k) hasIncome

(l) householdSize

(m) PE040

(n) PL031

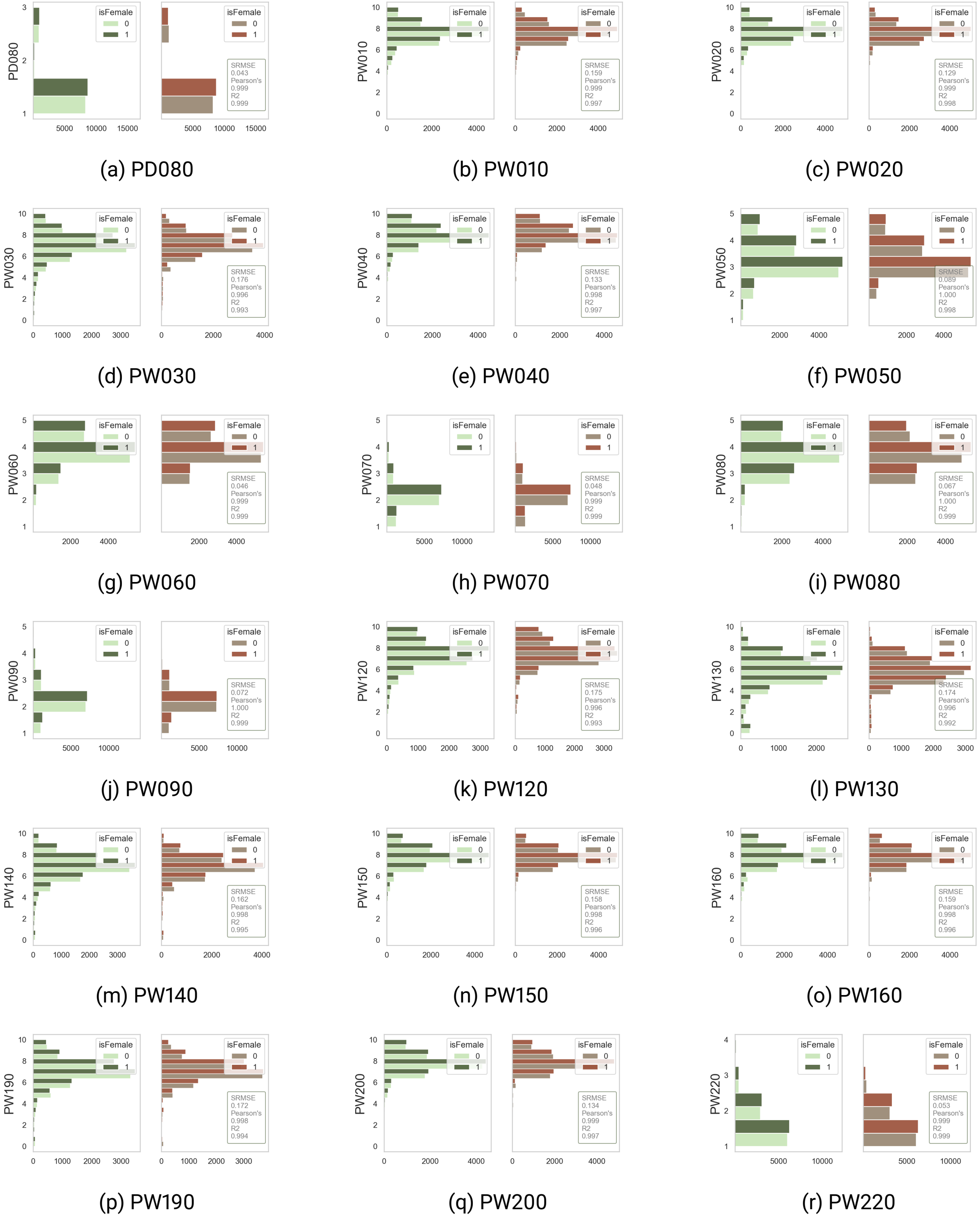
(o) PL040

(p) PD020

(q) PD050

(r) PD060

Figure 57: WGAN-50 marginals for all single variables from EU-SILC Finland visualised with second variable gender. Green bars are from original and red bars from synthetic data. Differences on gender is visualised, and show the relationship between two variables. Metrics shown in figure is for difference between original and synthetic main variable.



(a) PD080

(b) PW010

(c) PW020

(d) PW030

(e) PW040

(f) PW050

(g) PW060

(h) PW070

(i) PW080

(j) PW090

(k) PW120

(l) PW130

(m) PW140

(n) PW150

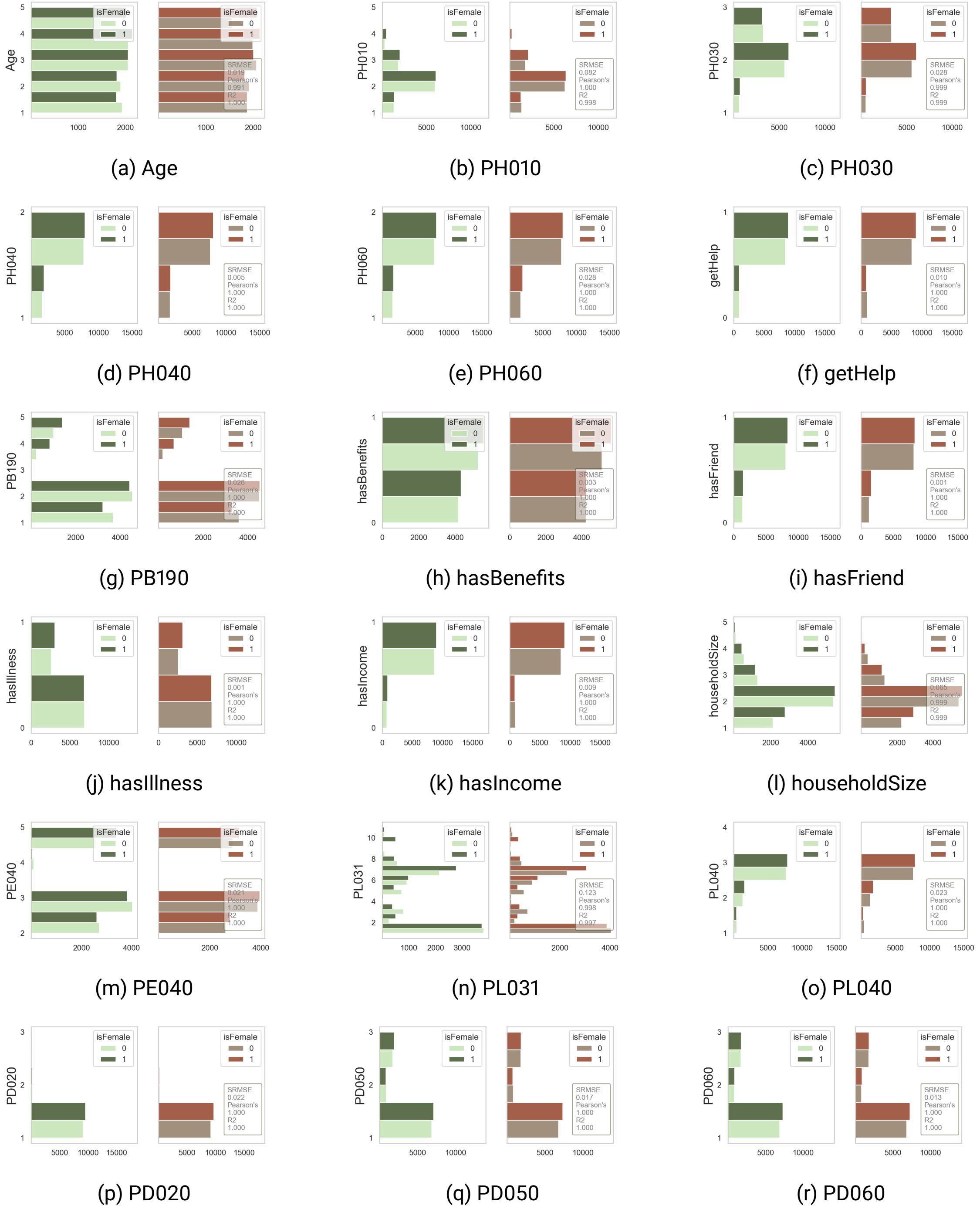
(o) PW160

(p) PW190

(q) PW200

(r) PW220

Figure 58: WGAN-50 marginals for all single variables from EU-SILC Finland visualised with second variable gender. Green bars are from original and red bars from synthetic data. Differences on gender is visualised, and show the relationship between two variables. Metrics shown in figure is for difference between original and synthetic main variable.



(a) Age

(b) PH010

(c) PH030

(d) PH040

(e) PH060

(f) getHelp

(g) PB190

(h) hasBenefits

(i) hasFriend

(j) hasIllness

(k) hasIncome

(l) householdSize

(m) PE040

(n) PL031

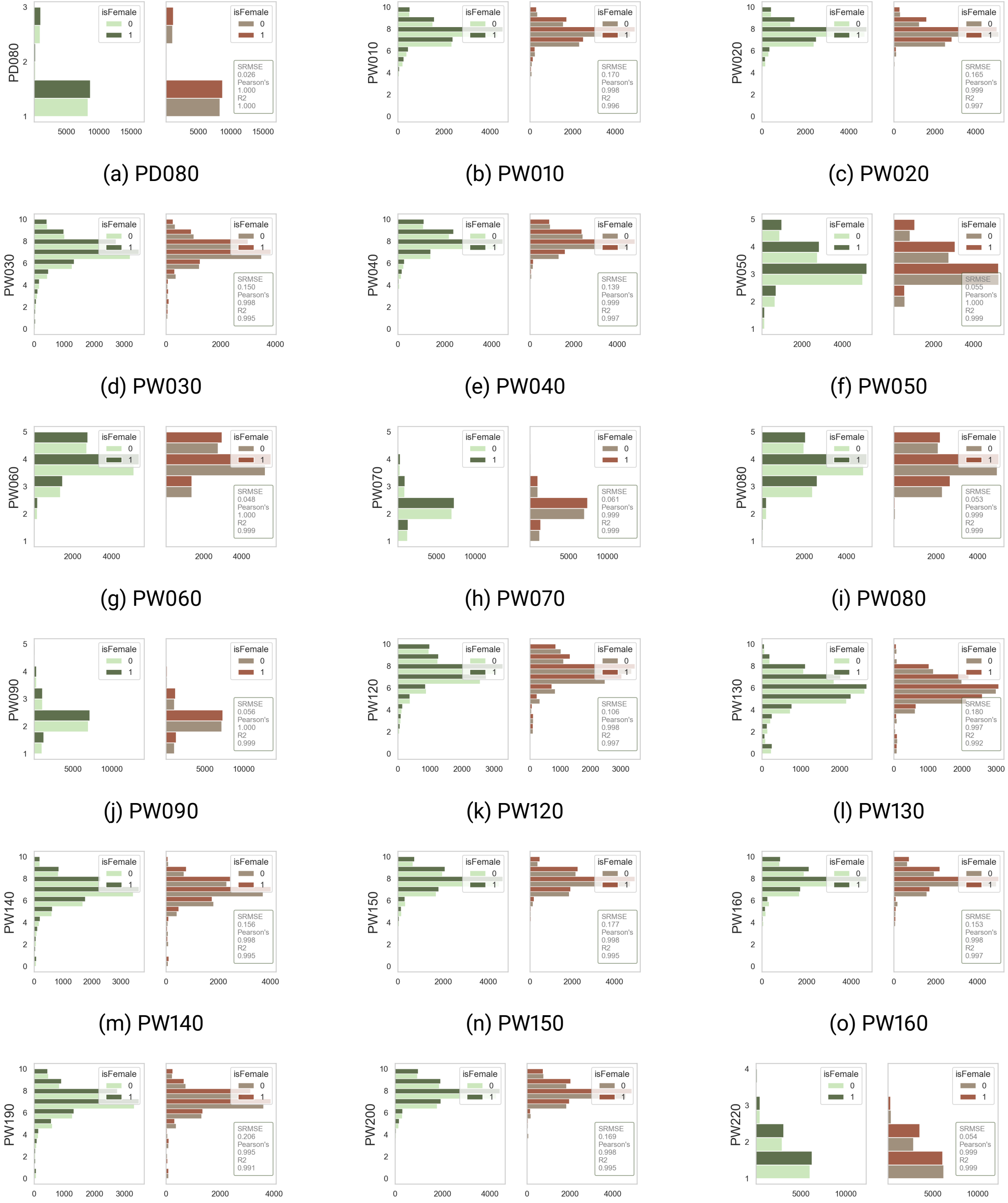
(o) PL040

(p) PD020

(q) PD050

(r) PD060

Figure 59: WGAN-100 marginals for all single variables from EU-SILC Finland visualised with second variable gender. Green bars are from original and red bars from synthetic data. Differences on gender is visualised, and show the relationship between two variables. Metrics shown in figure is for difference between original and synthetic main variable.



(a) PD080

(b) PW010

(c) PW020

(d) PW030

(e) PW040

(f) PW050

(g) PW060

(h) PW070

(i) PW080

(j) PW090

(k) PW120

(l) PW130

(m) PW140

(n) PW150

(o) PW160

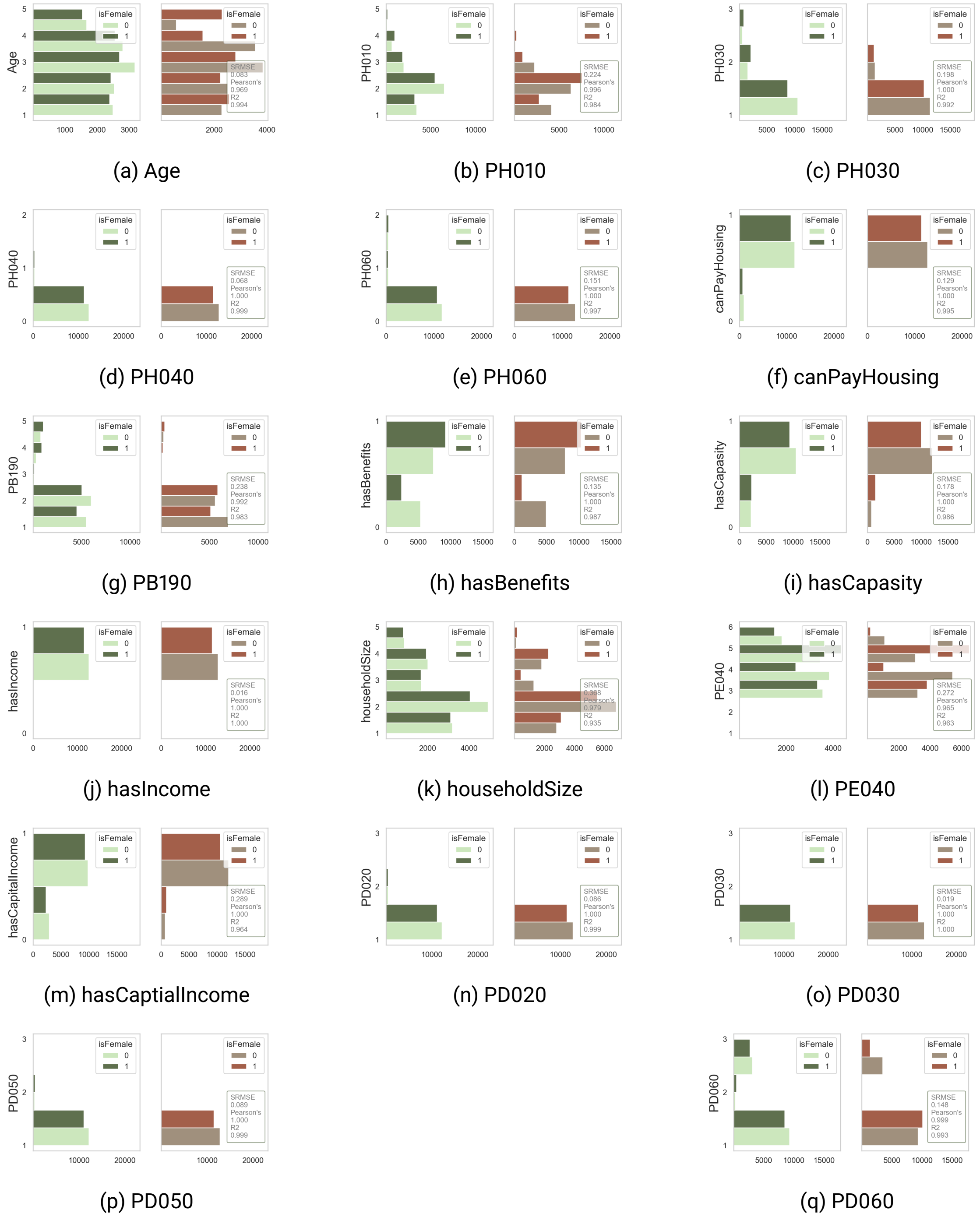
(p) PW190

(q) PW200

(r) PW220

Figure 60: WGAN-100 marginals for all single variables from EU-SILC Finland visualised with second variable gender. Green bars are from original and red bars from synthetic data. Differences on gender is visualised, and show the relationship between two variables. Metrics shown in figure is for difference between original and synthetic main variable.

F Single Variables EU-SILC Norway 2017-2020



(a) Age

(b) PH010

(c) PH030

(d) PH040

(e) PH060

(f) canPayHousing

(g) PB190

(h) hasBenefits

(i) hasCapasity

(j) hasIncome

(k) householdSize

(l) PE040

(m) hasCaptialIncome

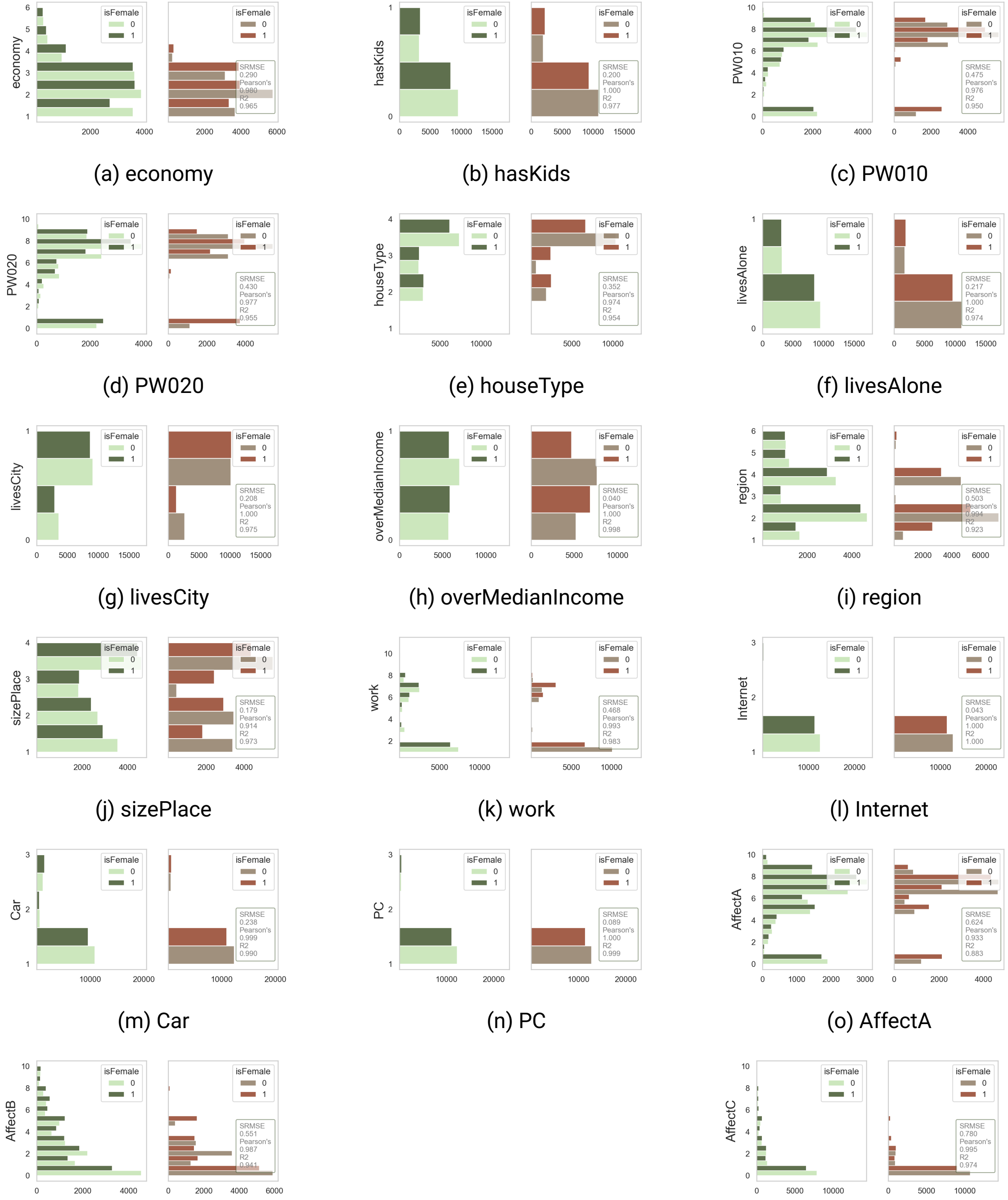
(n) PD020

(o) PD030

(p) PD050

(q) PD060

Figure 61: VAE-15 marginals for all single variables from EU-SILC Norway visualised with second variable gender. Green bars are from original and red bars from synthetic data. Differences on gender is visualised, and show the relationship between two variables. Metrics shown in figure is for difference between original and synthetic main variable.



(a) economy

(b) hasKids

(c) PW010

(d) PW020

(e) houseType

(f) livesAlone

(g) livesCity

(h) overMedianIncome

(i) region

(j) sizePlace

(k) work

(l) Internet

(m) Car

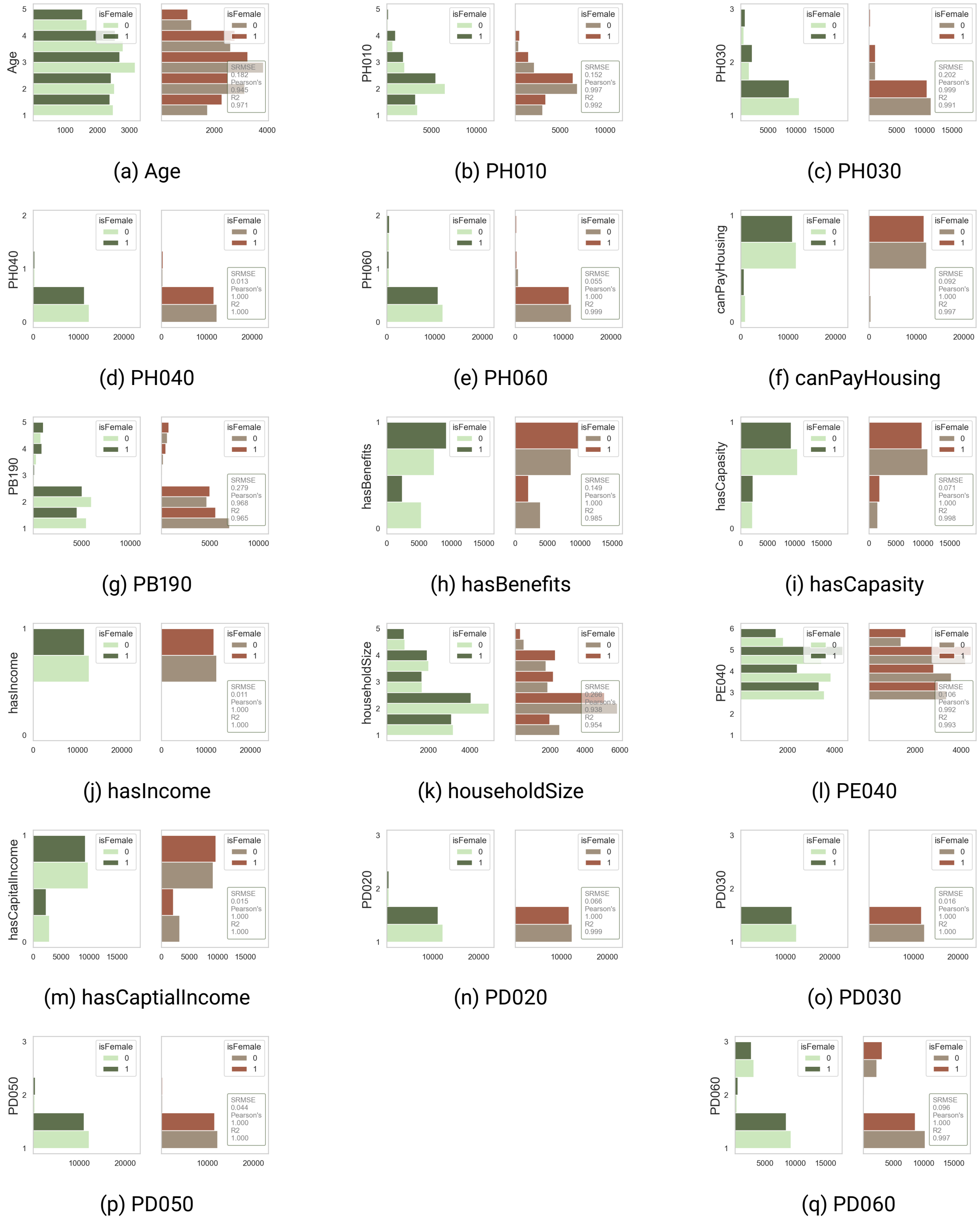
(n) PC

(o) AffectA

(p) AffectB

(q) AffectC

Figure 62: VAE-15 marginals for all single variables from EU-SILC Norway visualised with second variable gender. Green bars are from original and red bars from synthetic data. Differences on gender is visualised, and show the relationship between two variables. Metrics shown in figure is for difference between original and synthetic main variable.



(a) Age

(b) PH010

(c) PH030

(d) PH040

(e) PH060

(f) canPayHousing

(g) PB190

(h) hasBenefits

(i) hasCapasity

(j) hasIncome

(k) householdSize

(l) PE040

(m) hasCaptialIncome

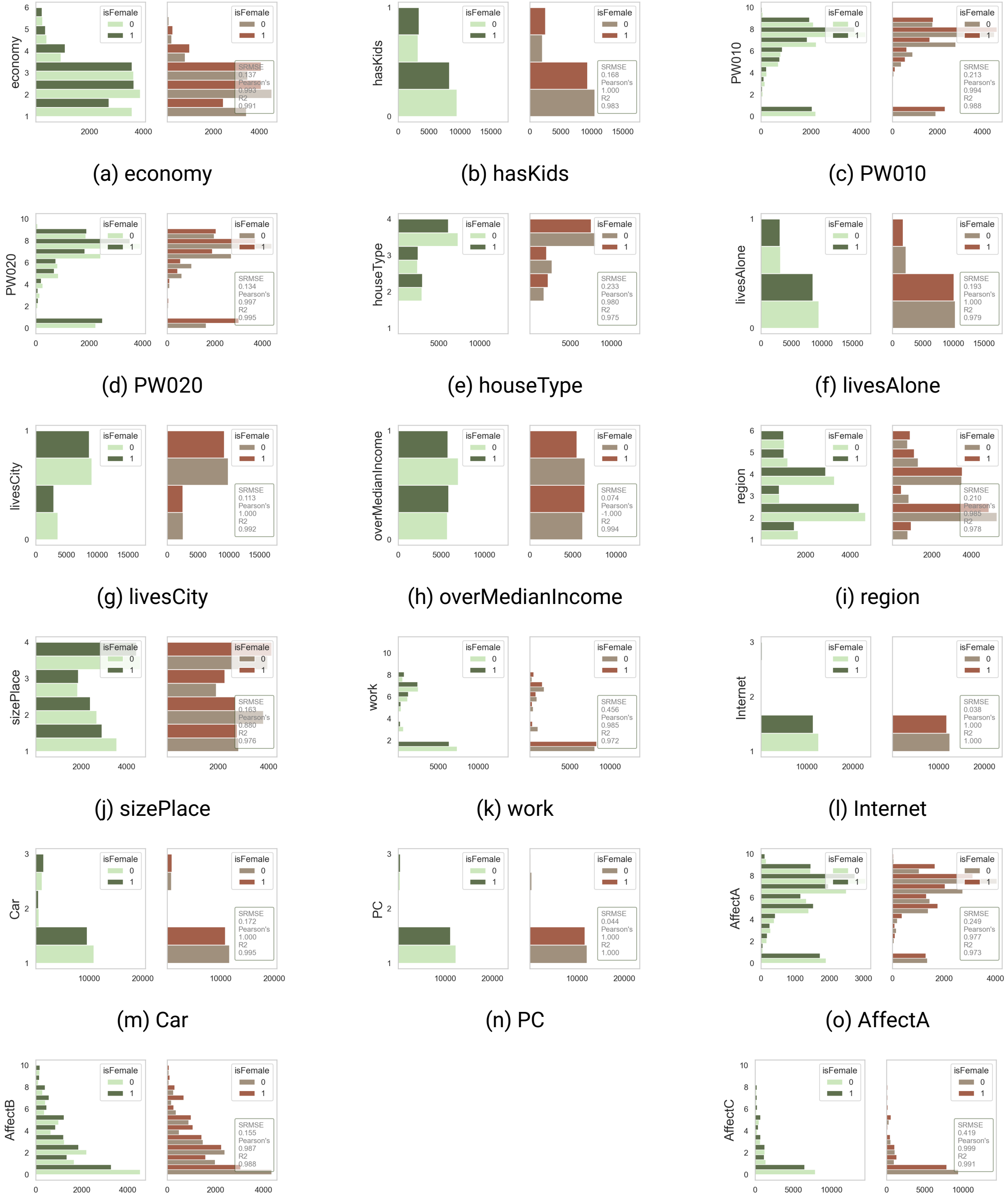
(n) PD020

(o) PD030

(p) PD050

(q) PD060

Figure 63: VAE-30 marginals for all single variables from EU-SILC Norway visualised with second variable gender. Green bars are from original and red bars from synthetic data. Differences on gender is visualised, and show the relationship between two variables. Metrics shown in figure is for difference between original and synthetic main variable.



(a) economy

(b) hasKids

(c) PW010

(d) PW020

(e) houseType

(f) livesAlone

(g) livesCity

(h) overMedianIncome

(i) region

(j) sizePlace

(k) work

(l) Internet

(m) Car

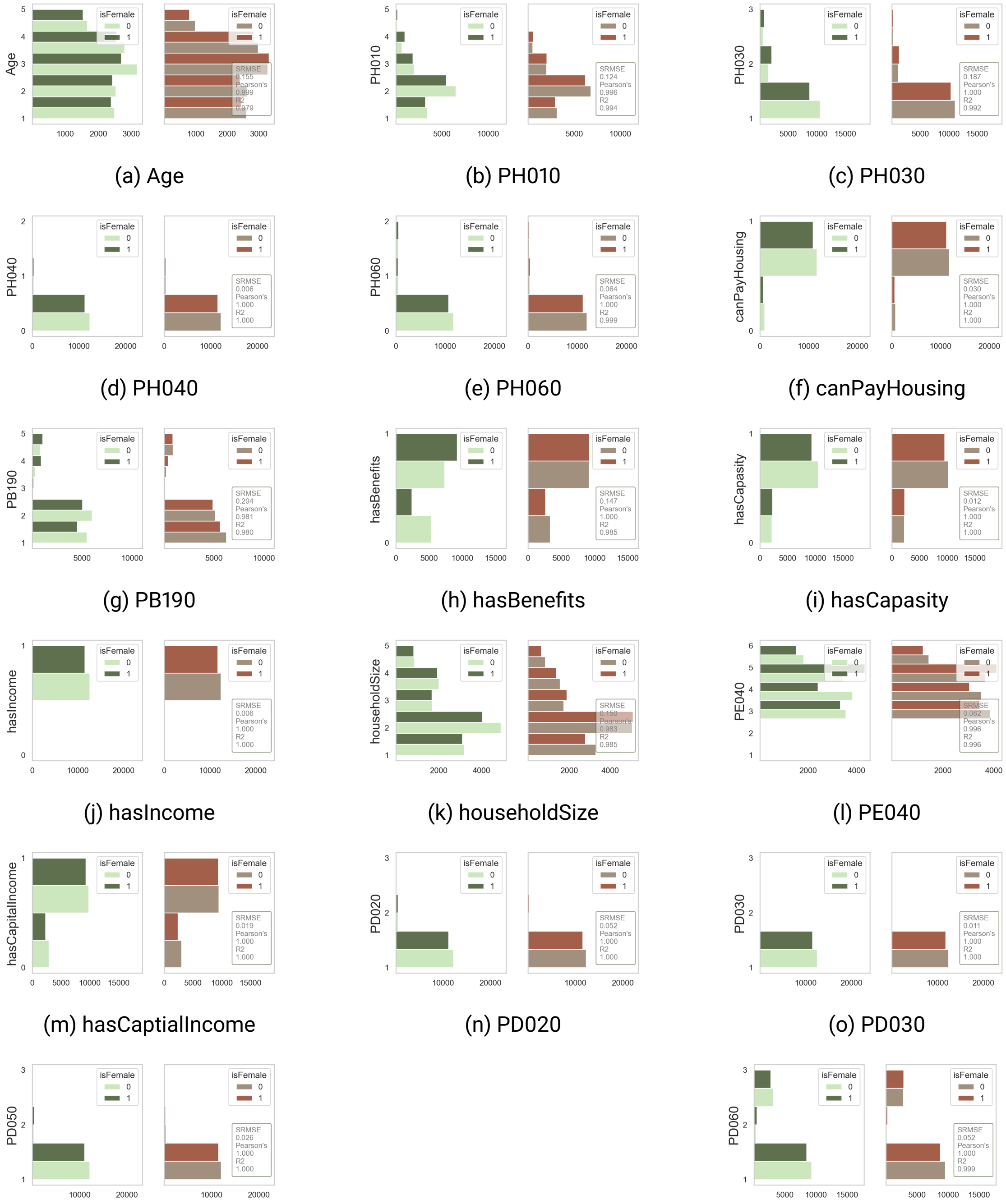
(n) PC

(o) AffectA

(p) AffectB

(q) AffectC

Figure 64: VAE-30 marginals for all single variables from EU-SILC Norway visualised with second variable gender. Green bars are from original and red bars from synthetic data. Differences on gender is visualised, and show the relationship between two variables. Metrics shown in figure is for difference between original and synthetic main variable.



(a) Age

(b) PH010

(c) PH030

(d) PH040

(e) PH060

(f) canPayHousing

(g) PB190

(h) hasBenefits

(i) hasCapasity

(j) hasIncome

(k) householdSize

(l) PE040

(m) hasCaptialIncome

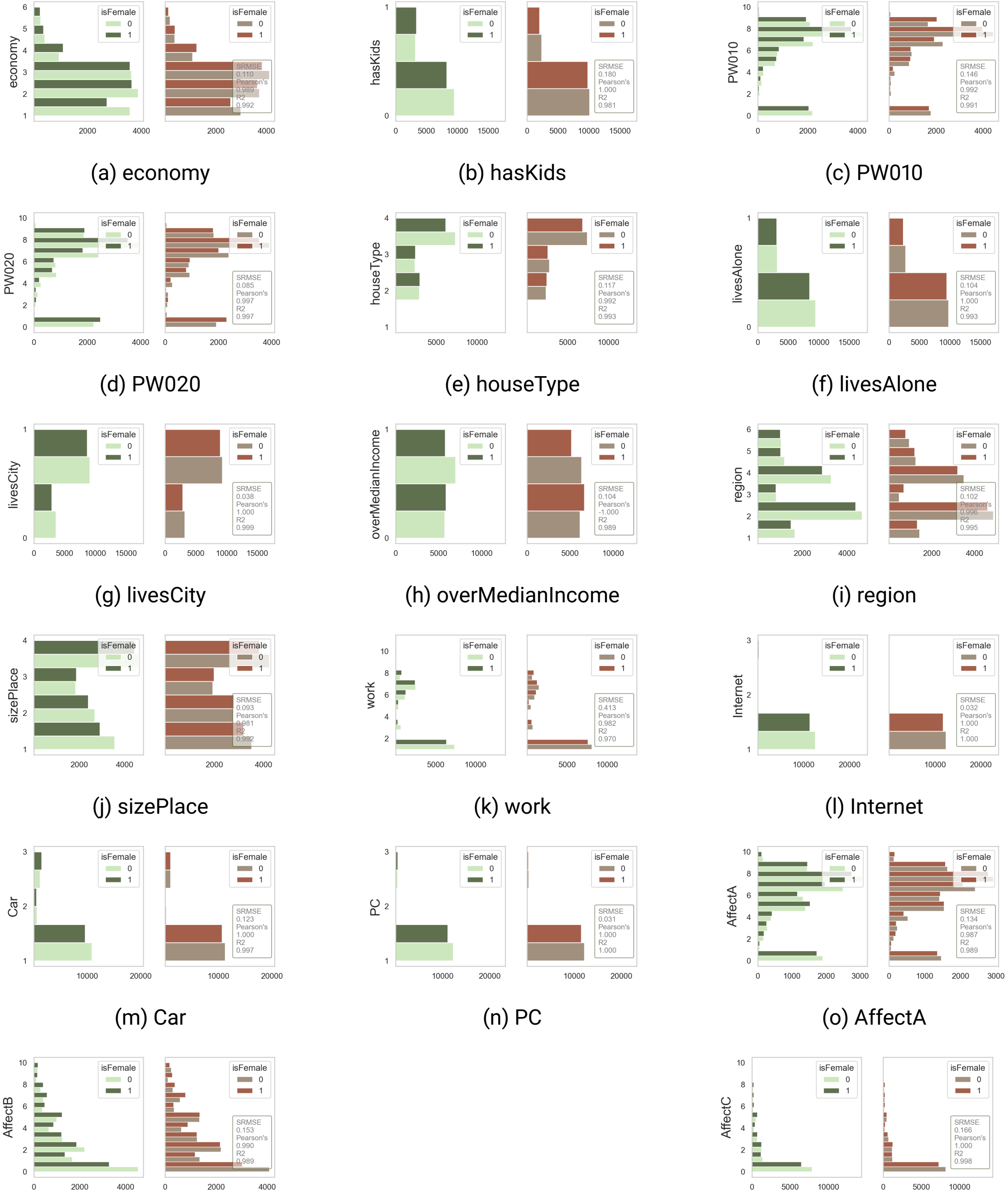
(n) PD020

(o) PD030

(p) PD050

(q) PD060

Figure 65: VAE-50 marginals for all single variables from EU-SILC Norway visualised with second variable gender. Green bars are from original and red bars from synthetic data. Differences on gender is visualised, and show the relationship between two variables. Metrics shown in figure is for difference between original and synthetic main variable.



(a) economy

(b) hasKids

(c) PW010

(d) PW020

(e) houseType

(f) livesAlone

(g) livesCity

(h) overMedianIncome

(i) region

(j) sizePlace

(k) work

(l) Internet

(m) Car

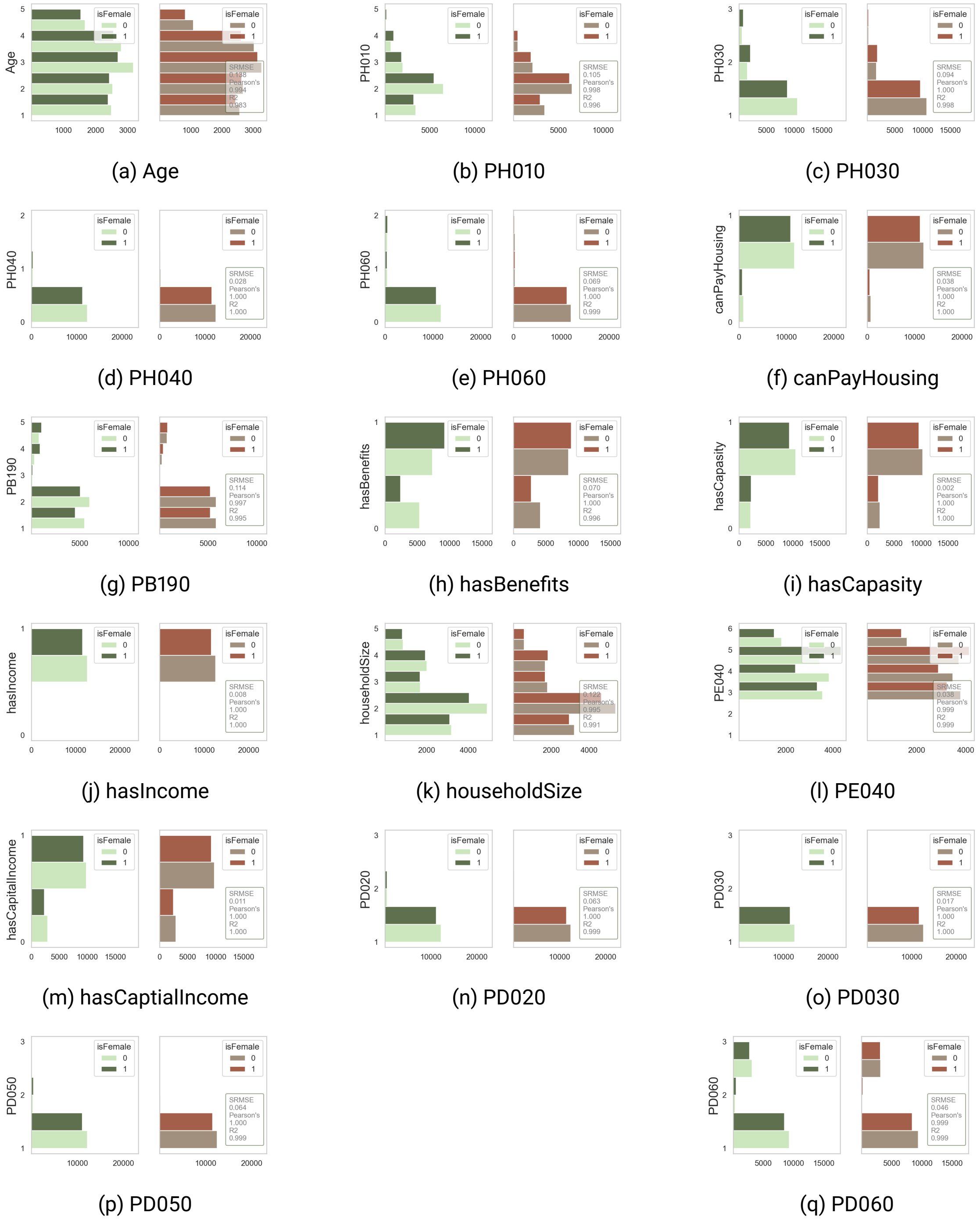
(n) PC

(o) AffectA

(p) AffectB

(q) AffectC

Figure 66: VAE-50 marginals for all single variables from EU-SILC Norway visualised with second variable gender. Green bars are from original and red bars from synthetic data. Differences on gender is visualised, and show the relationship between two variables. Metrics shown in figure is for difference between original and synthetic main variable.



(a) Age

(b) PH010

(c) PH030

(d) PH040

(e) PH060

(f) canPayHousing

(g) PB190

(h) hasBenefits

(i) hasCapasity

(j) hasIncome

(k) householdSize

(l) PE040

(m) hasCaptialIncome

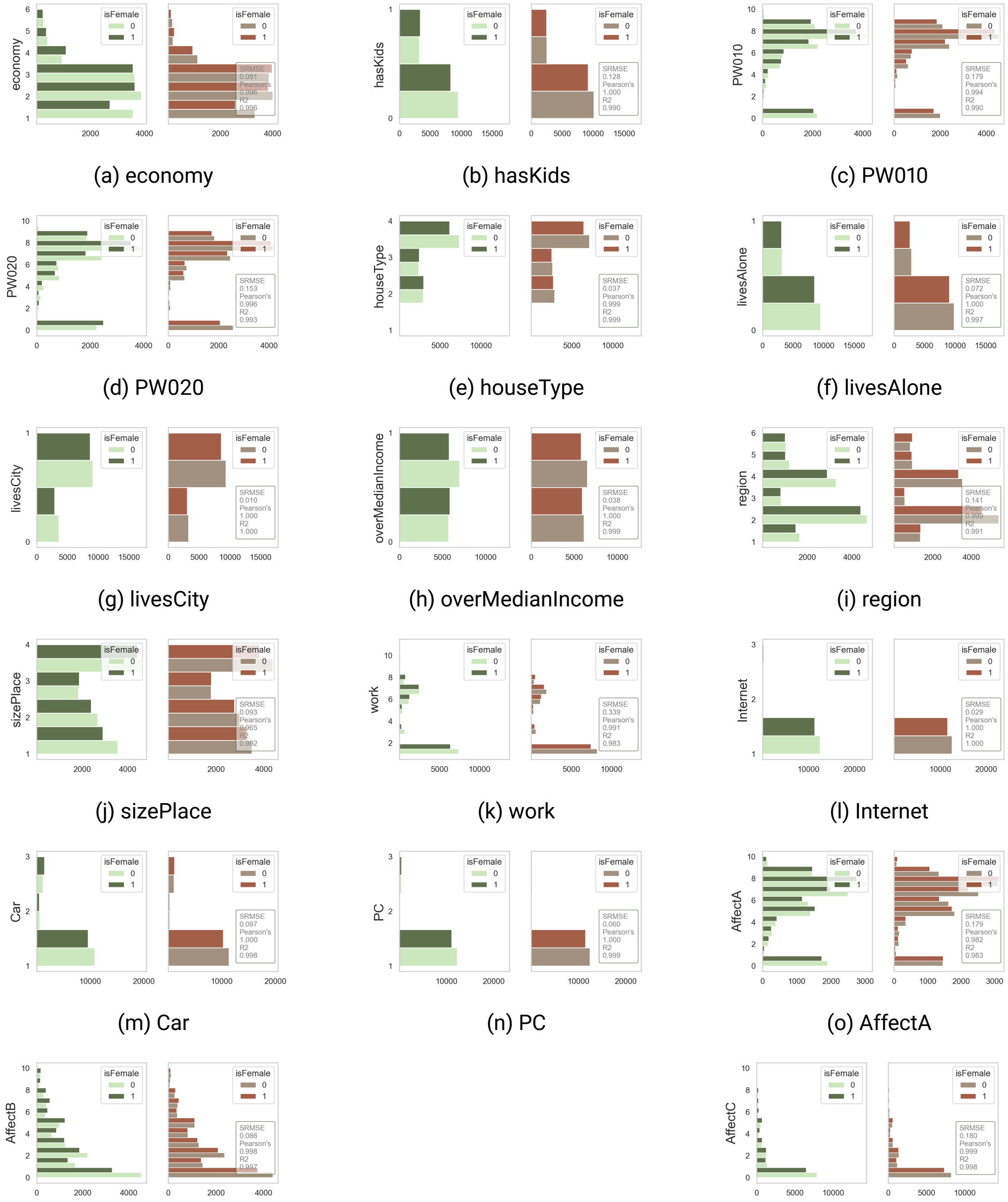
(n) PD020

(o) PD030

(p) PD050

(q) PD060

Figure 67: VAE-100 marginals for all single variables from EU-SILC Norway visualised with second variable gender. Green bars are from original and red bars from synthetic data. Differences on gender is visualised, and show the relationship between two variables. Metrics shown in figure is for difference between original and synthetic main variable.



(a) economy

(b) hasKids

(c) PW010

(d) PW020

(e) houseType

(f) livesAlone

(g) livesCity

(h) overMedianIncome

(i) region

(j) sizePlace

(k) work

(l) Internet

(m) Car

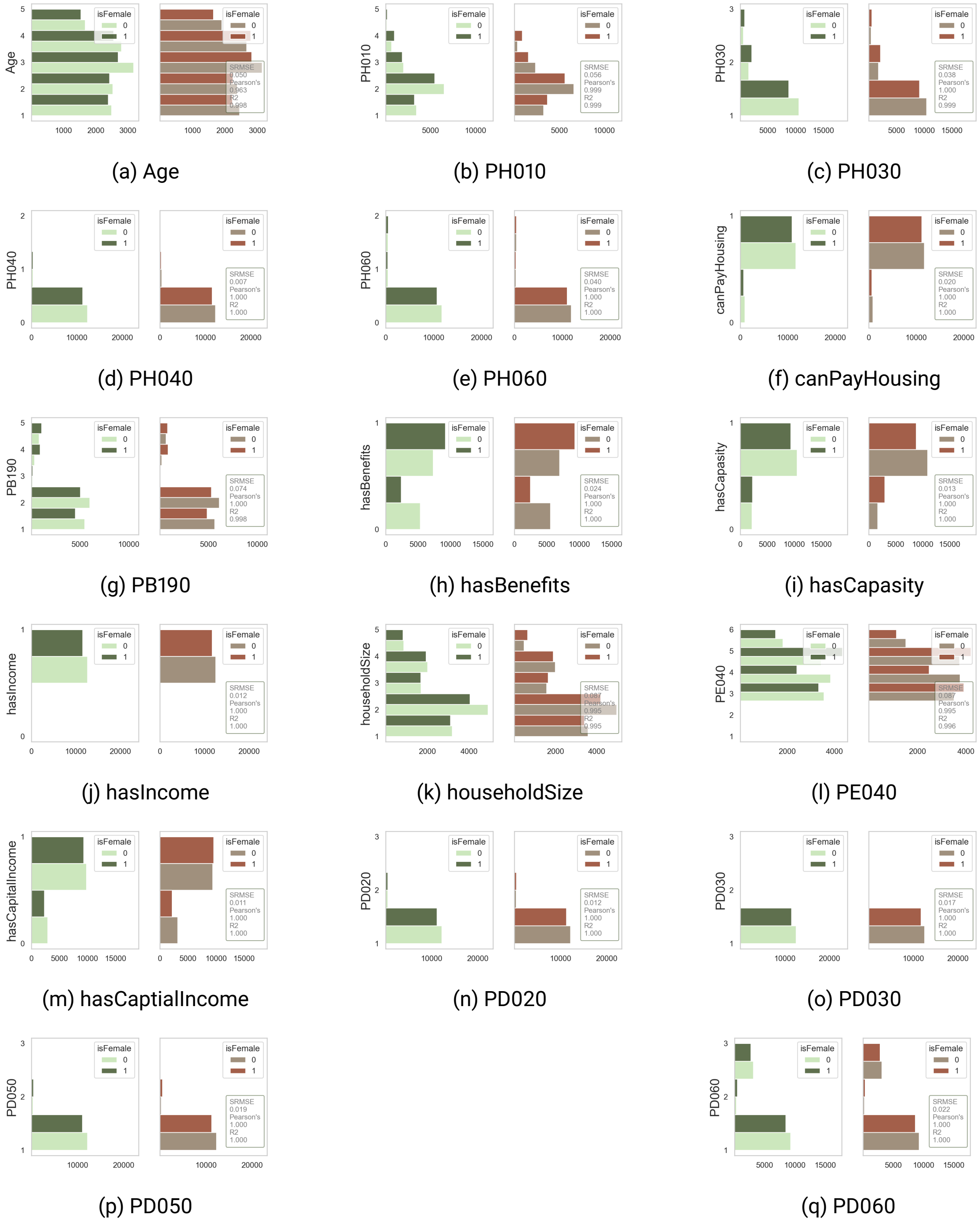
(n) PC

(o) AffectA

(p) AffectB

(q) AffectC

Figure 68: VAE-100 marginals for all single variables from EU-SILC Norway visualised with second variable gender. Green bars are from original and red bars from synthetic data. Differences on gender is visualised, and show the relationship between two variables. Metrics shown in figure is for difference between original and synthetic main variable.



(a) Age

(b) PH010

(c) PH030

(d) PH040

(e) PH060

(f) canPayHousing

(g) PB190

(h) hasBenefits

(i) hasCapasity

(j) hasIncome

(k) householdSize

(l) PE040

(m) hasCaptialIncome

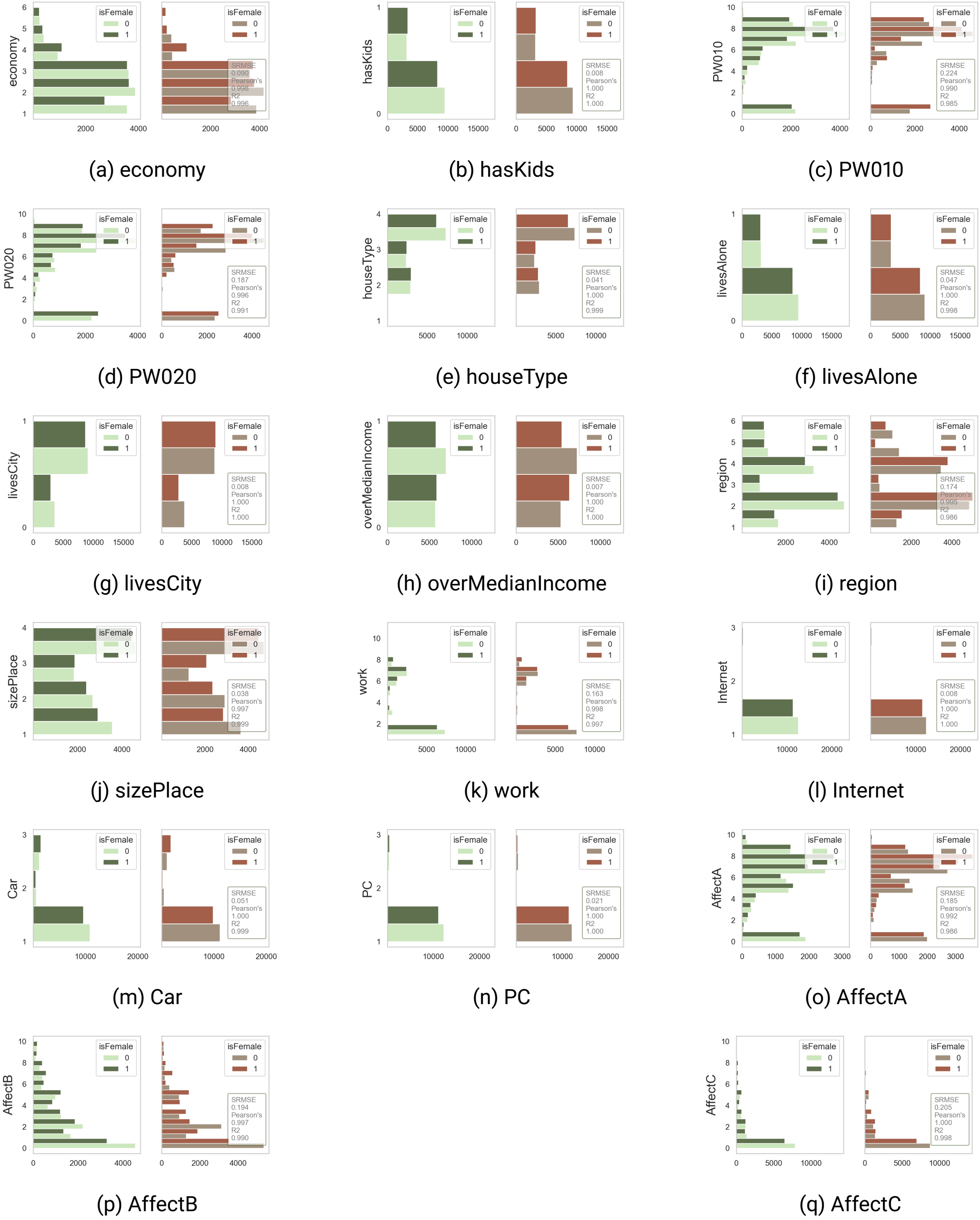
(n) PD020

(o) PD030

(p) PD050

(q) PD060

Figure 69: WGAN-15 marginals for all single variables from EU-SILC Norway visualised with second variable gender. Green bars are from original and red bars from synthetic data. Differences on gender is visualised, and show the relationship between two variables. Metrics shown in figure is for difference between original and synthetic main variable.



(a) economy

(b) hasKids

(c) PW010

(d) PW020

(e) houseType

(f) livesAlone

(g) livesCity

(h) overMedianIncome

(i) region

(j) sizePlace

(k) work

(l) Internet

(m) Car

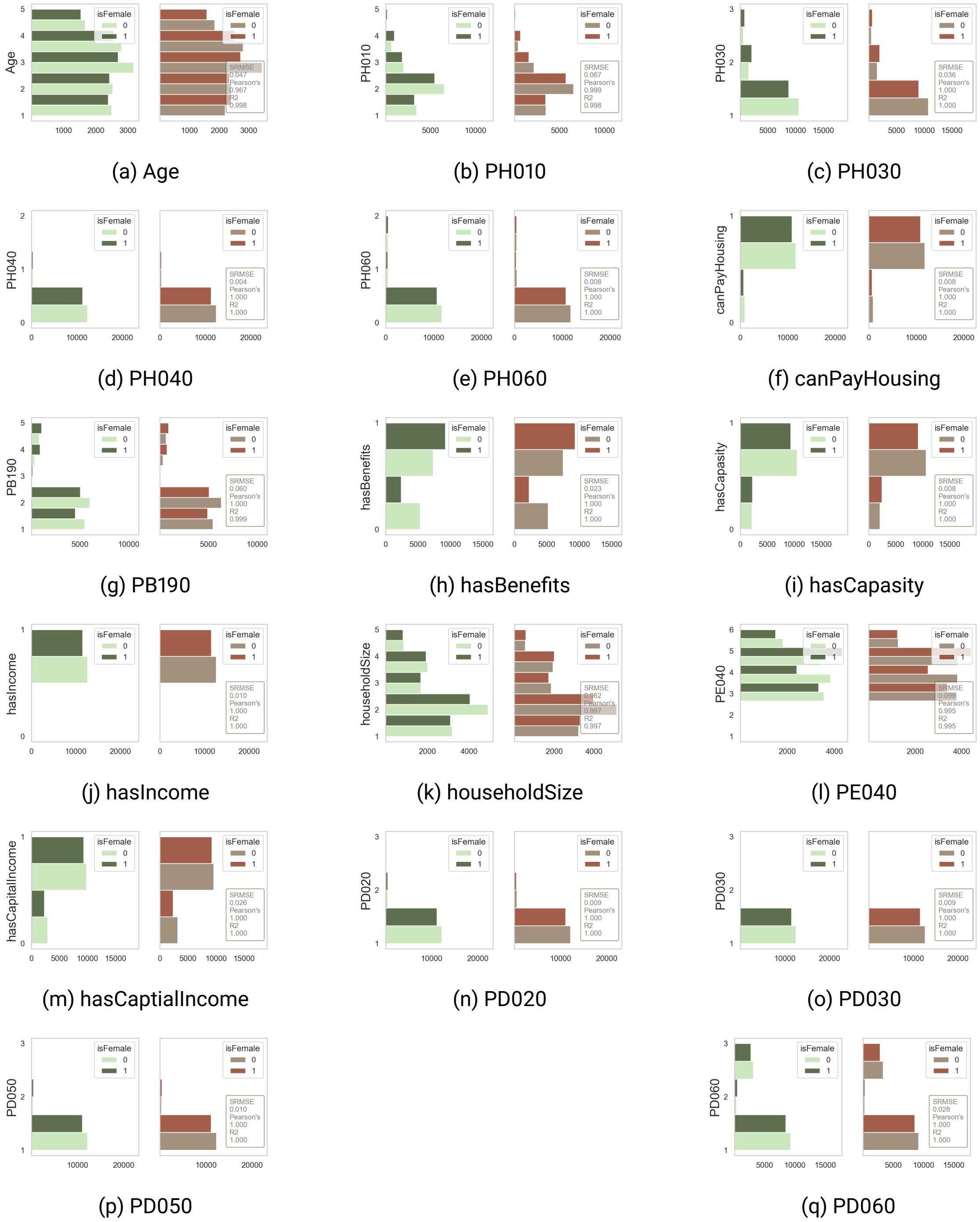
(n) PC

(o) AffectA

(p) AffectB

(q) AffectC

Figure 70: WGAN-15 marginals for all single variables from EU-SILC Norway visualised with second variable gender. Green bars are from original and red bars from synthetic data. Differences on gender is visualised, and show the relationship between two variables. Metrics shown in figure is for difference between original and synthetic main variable.



(a) Age

(b) PH010

(c) PH030

(d) PH040

(e) PH060

(f) canPayHousing

(g) PB190

(h) hasBenefits

(i) hasCapasity

(j) hasIncome

(k) householdSize

(l) PE040

(m) hasCaptialIncome

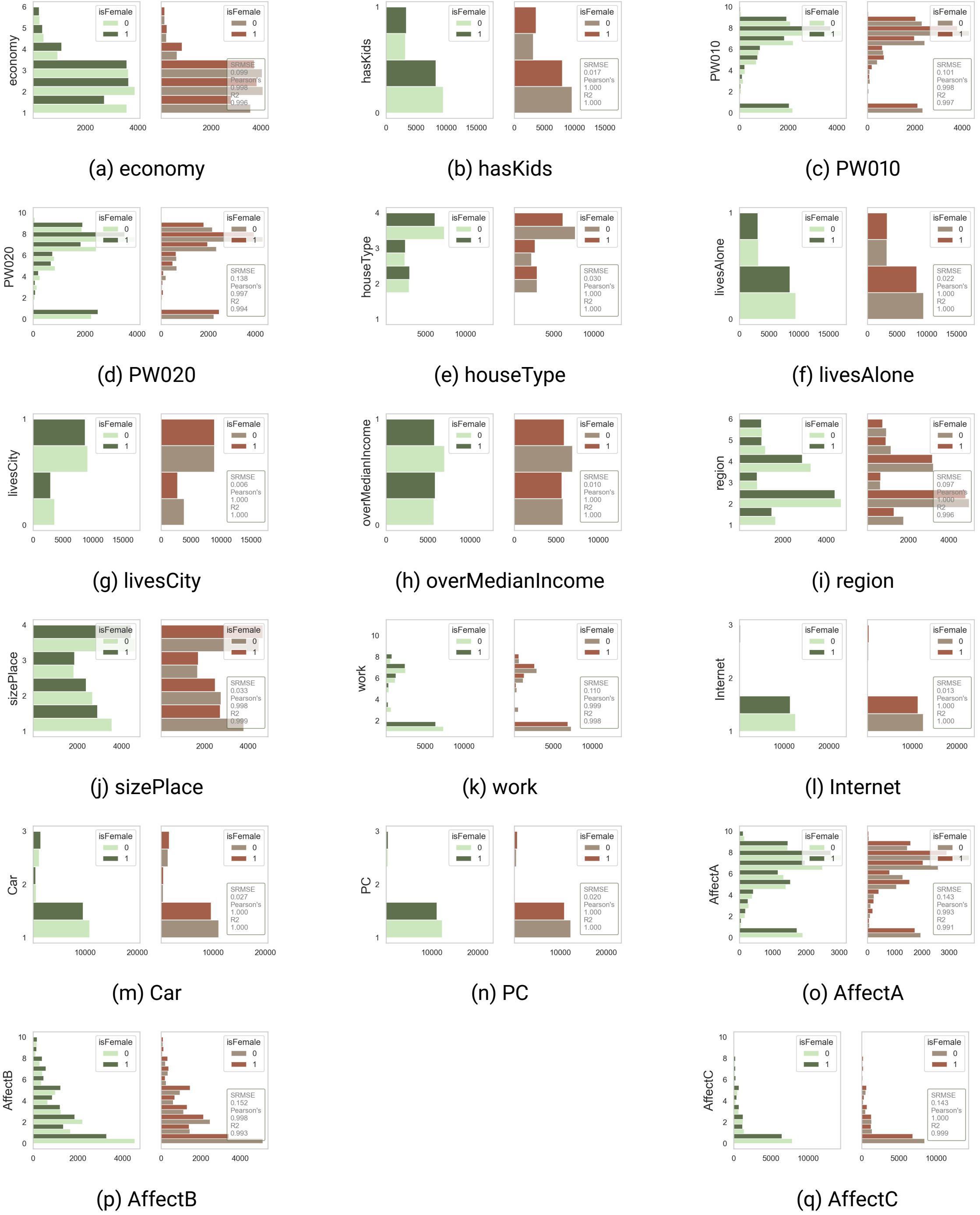
(n) PD020

(o) PD030

(p) PD050

(q) PD060

Figure 71: WGAN-30 marginals for all single variables from EU-SILC Norway visualised with second variable gender. Green bars are from original and red bars from synthetic data. Differences on gender is visualised, and show the relationship between two variables. Metrics shown in figure is for difference between original and synthetic main variable.



(a) economy

(b) hasKids

(c) PW010

(d) PW020

(e) houseType

(f) livesAlone

(g) livesCity

(h) overMedianIncome

(i) region

(j) sizePlace

(k) work

(l) Internet

(m) Car

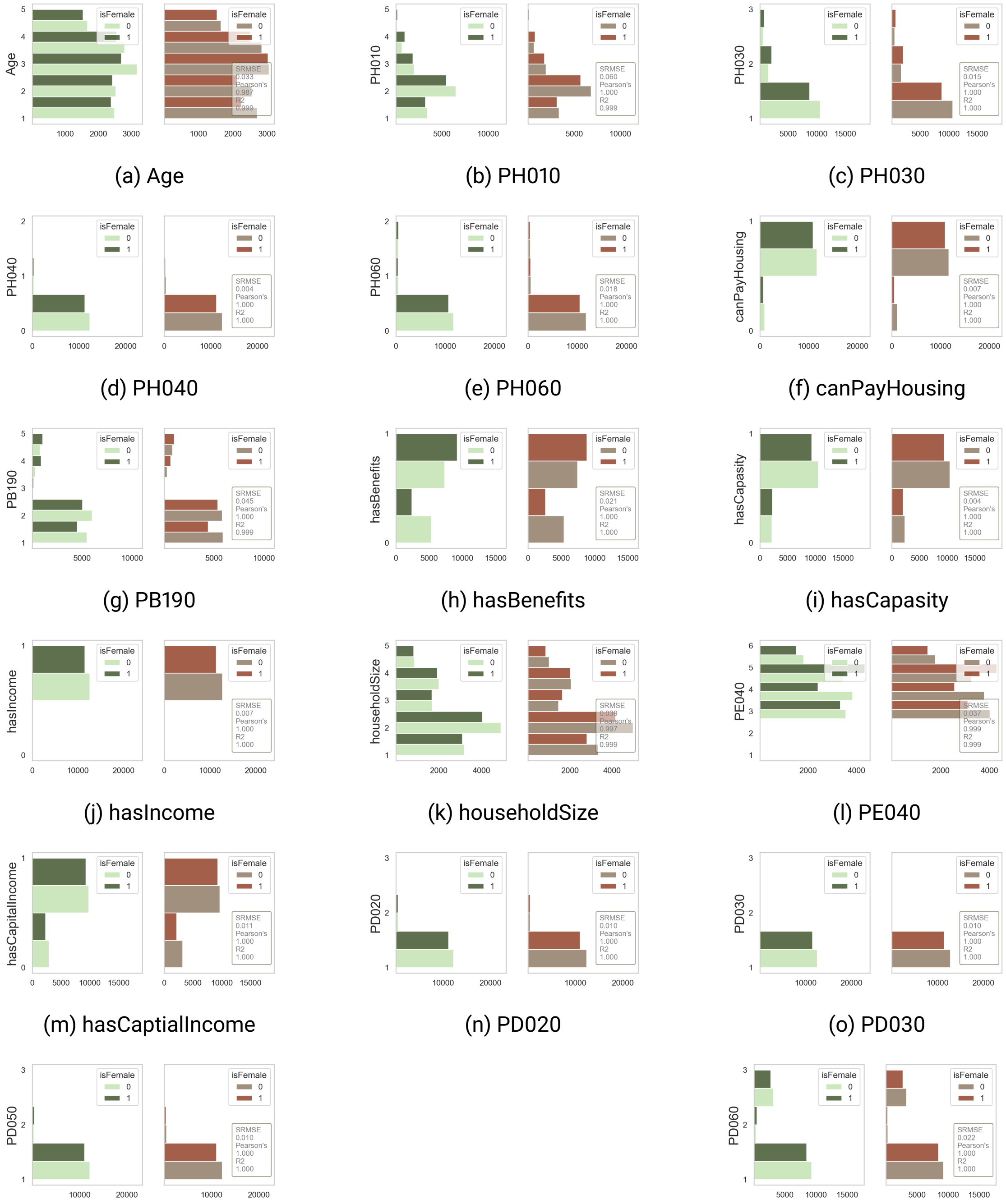
(n) PC

(o) AffectA

(p) AffectB

(q) AffectC

Figure 72: WGAN-30 marginals for all single variables from EU-SILC Norway visualised with second variable gender. Green bars are from original and red bars from synthetic data. Differences on gender is visualised, and show the relationship between two variables. Metrics shown in figure is for difference between original and synthetic main variable.



(a) Age

(b) PH010

(c) PH030

(d) PH040

(e) PH060

(f) canPayHousing

(g) PB190

(h) hasBenefits

(i) hasCapasity

(j) hasIncome

(k) householdSize

(l) PE040

(m) hasCaptialIncome

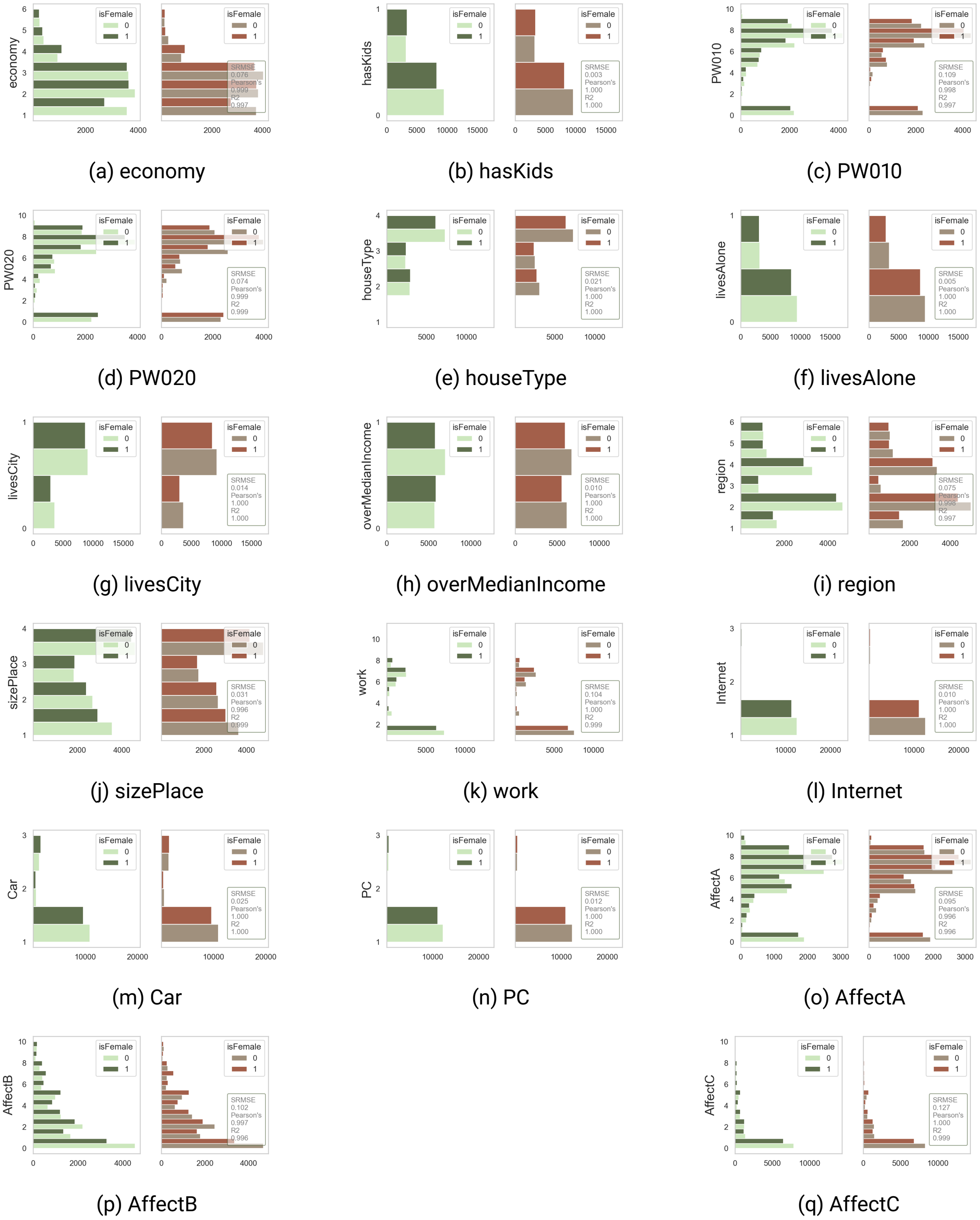
(n) PD020

(o) PD030

(p) PD050

(q) PD060

Figure 73: WGAN-50 marginals for all single variables from EU-SILC Norway visualised with second variable gender. Green bars are from original and red bars from synthetic data. Differences on gender is visualised, and show the relationship between two variables. Metrics shown in figure is for difference between original and synthetic main variable.



(a) economy

(b) hasKids

(c) PW010

(d) PW020

(e) houseType

(f) livesAlone

(g) livesCity

(h) overMedianIncome

(i) region

(j) sizePlace

(k) work

(l) Internet

(m) Car

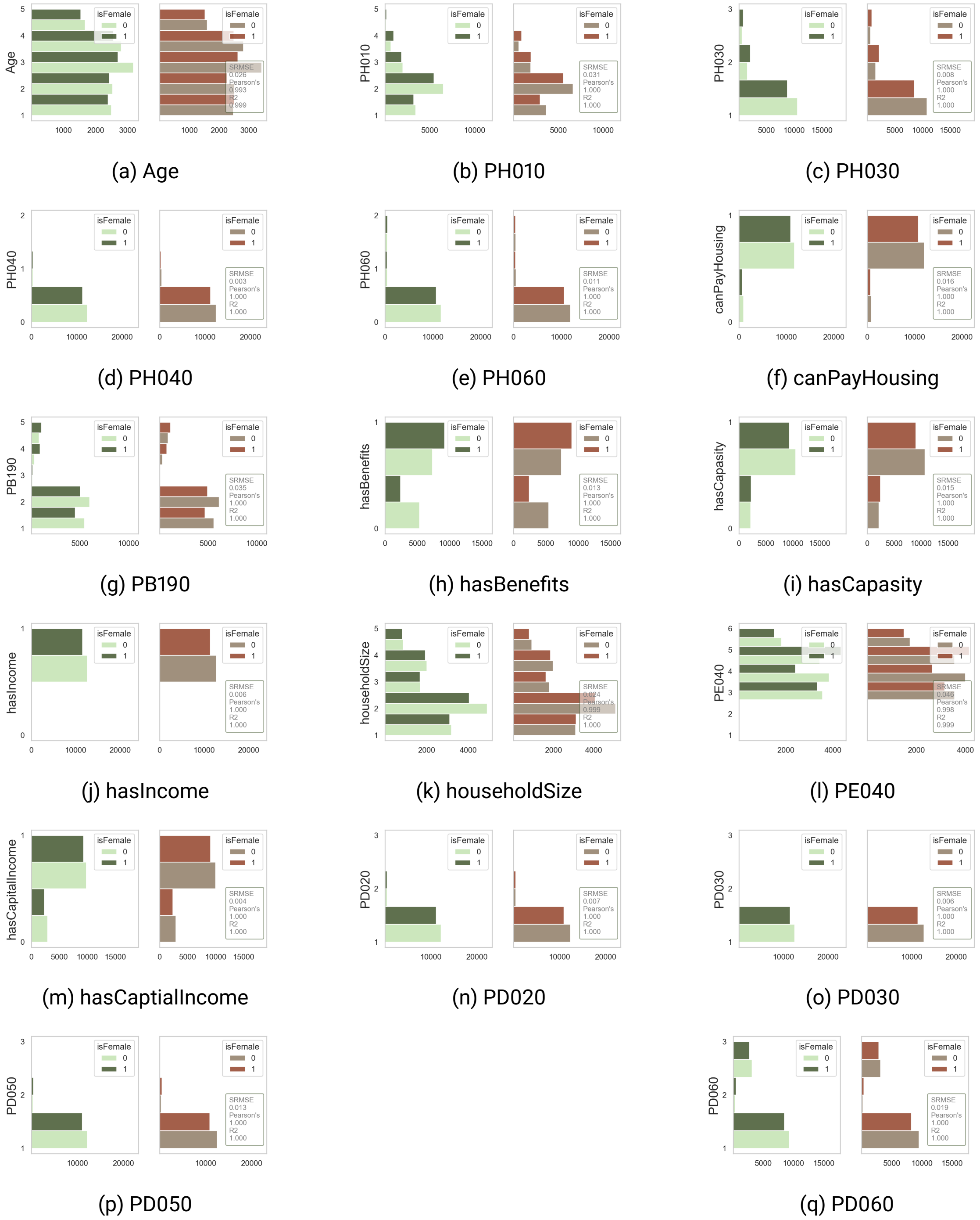
(n) PC

(o) AffectA

(p) AffectB

(q) AffectC

Figure 74: WGAN-50 marginals for all single variables from EU-SILC Norway visualised with second variable gender. Green bars are from original and red bars from synthetic data. Differences on gender is visualised, and show the relationship between two variables. Metrics shown in figure is for difference between original and synthetic main variable.



(a) Age

(b) PH010

(c) PH030

(d) PH040

(e) PH060

(f) canPayHousing

(g) PB190

(h) hasBenefits

(i) hasCapasity

(j) hasIncome

(k) householdSize

(l) PE040

(m) hasCaptialIncome

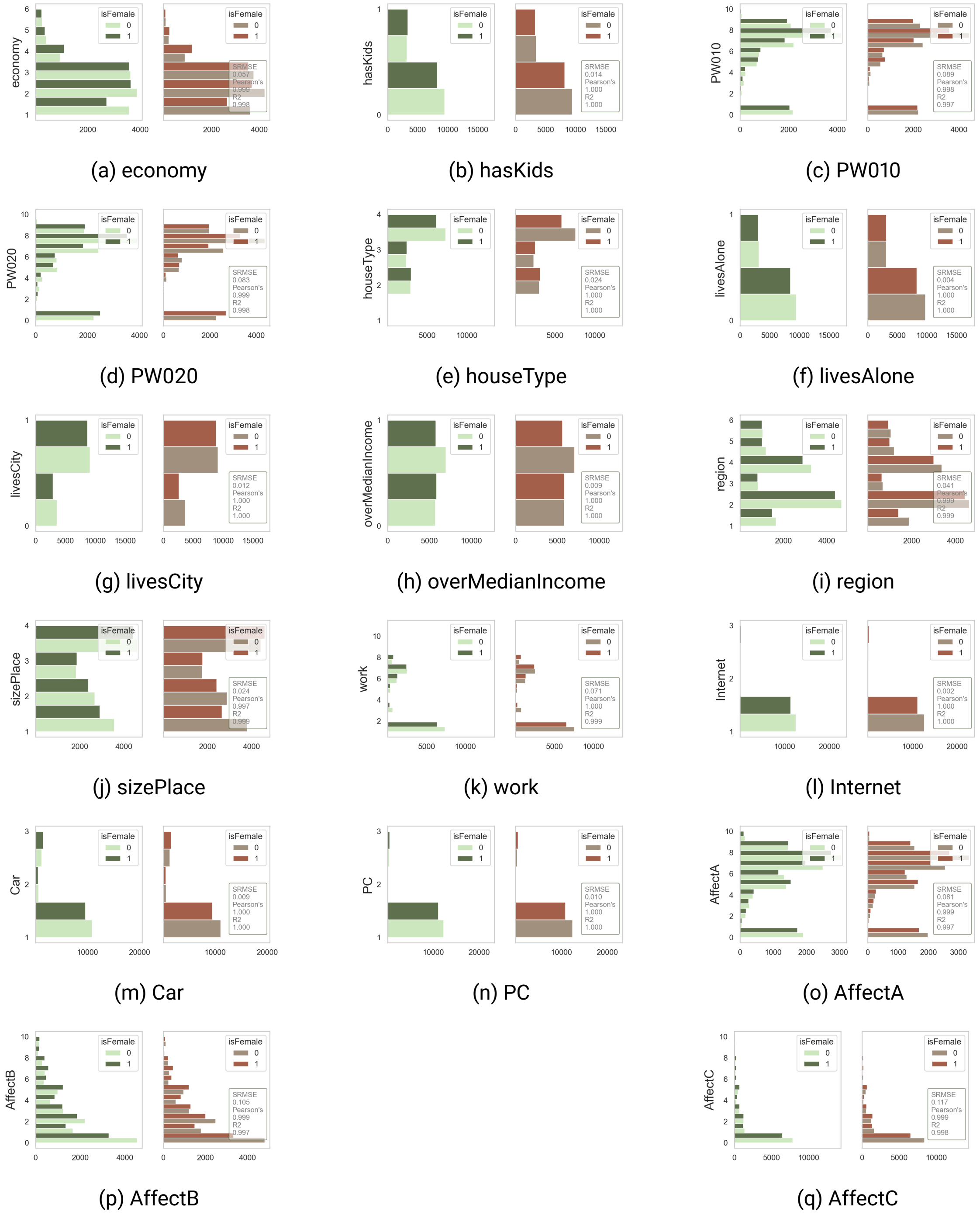
(n) PD020

(o) PD030

(p) PD050

(q) PD060

Figure 75: WGAN-100 marginals for all single variables from EU-SILC Norway visualised with second variable gender. Green bars are from original and red bars from synthetic data. Differences on gender is visualised, and show the relationship between two variables. Metrics shown in figure is for difference between original and synthetic main variable.



(a) economy

(b) hasKids

(c) PW010

(d) PW020

(e) houseType

(f) livesAlone

(g) livesCity

(h) overMedianIncome

(i) region

(j) sizePlace

(k) work

(l) Internet

(m) Car

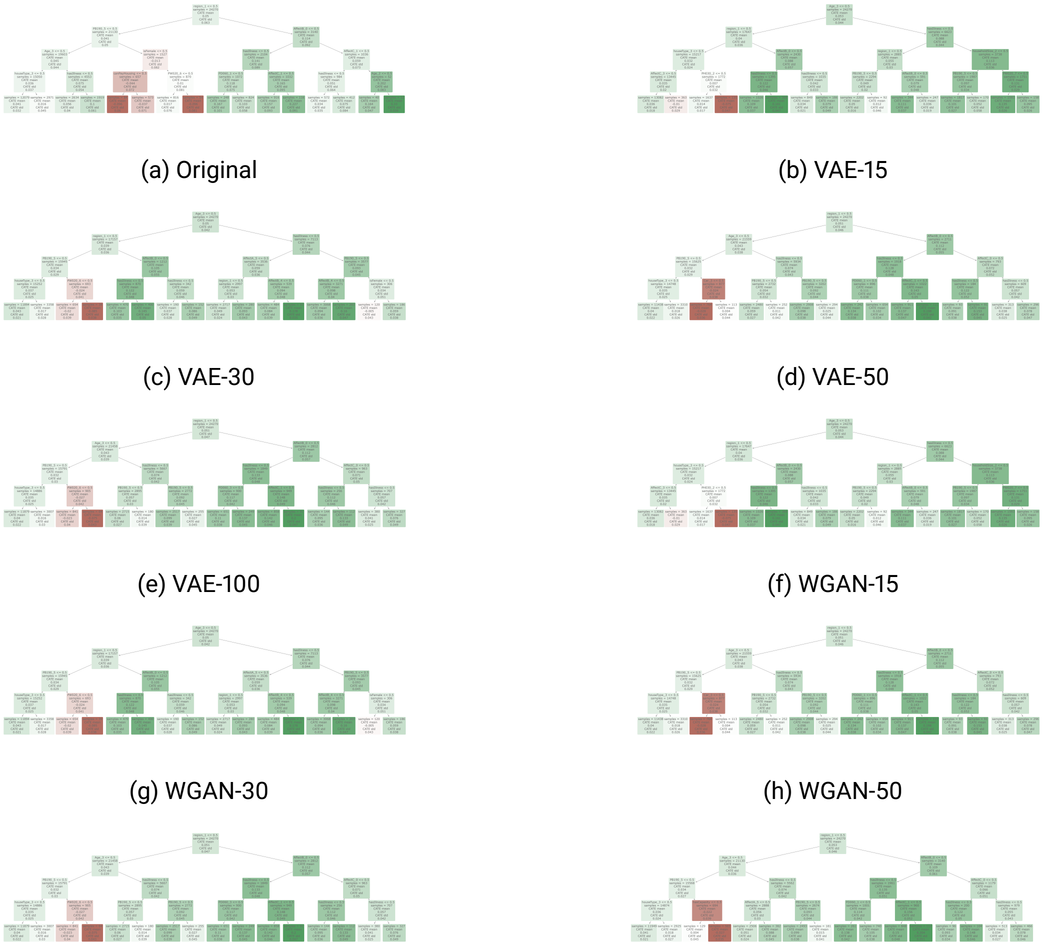
(n) PC

(o) AffectA

(p) AffectB

(q) AffectC

Figure 76: WGAN-100 marginals for all single variables from EU-SILC Norway visualised with second variable gender. Green bars are from original and red bars from synthetic data. Differences on gender is visualised, and show the relationship between two variables. Metrics shown in figure is for difference between original and synthetic main variable.



(a) Original

(b) VAE-15

(c) VAE-30

(d) VAE-50

(e) VAE-100

(f) WGAN-15

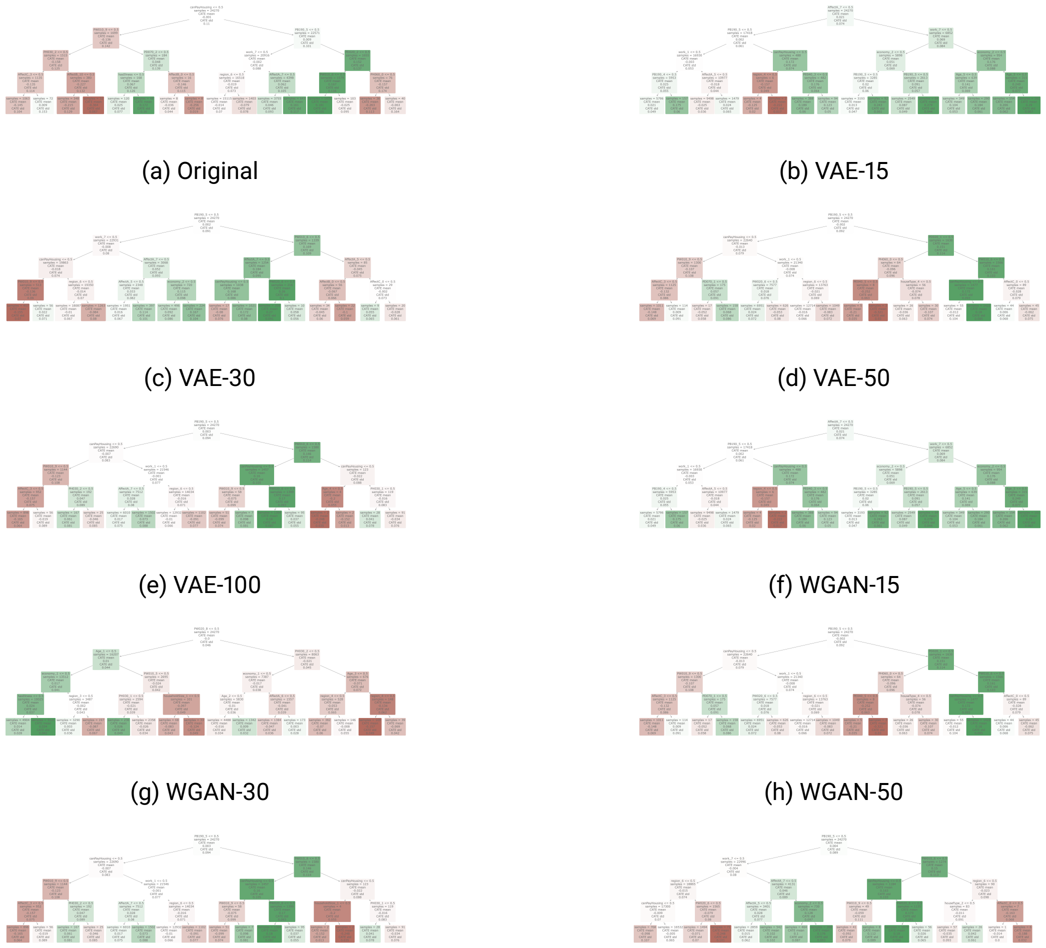
(g) WGAN-30

(h) WGAN-50

(i) WGAN-100

(j) Merged

Figure 77: Decision charts from Causal Forets DML run on intervention ”education B” on data from EU-SILC Norway, showing all deep generative models and the synthetic population resulting from generating regions by VAE-100 separately and then merge them into a population.



(a) Original

(b) VAE-15

(c) VAE-30

(d) VAE-50

(e) VAE-100

(f) WGAN-15

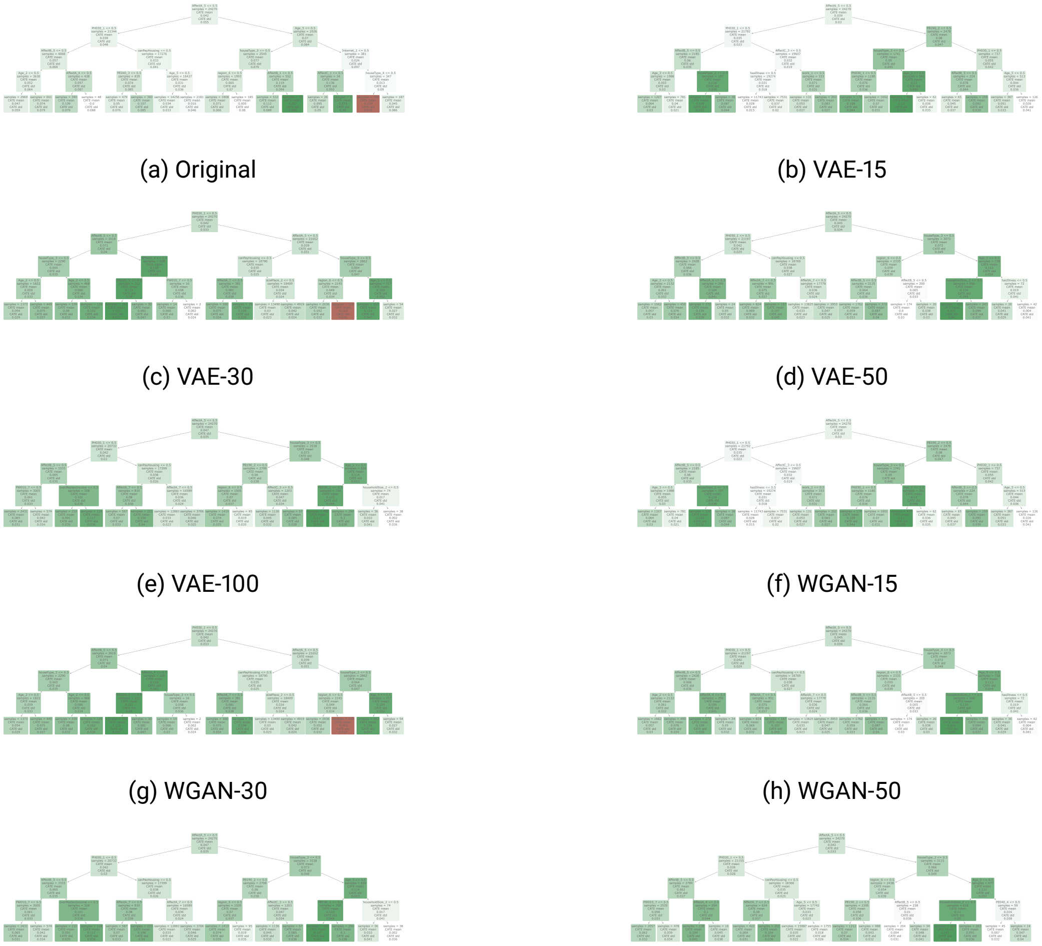
(g) WGAN-30

(h) WGAN-50

(i) WGAN-100

(j) Merged

Figure 78: Decision charts from Causal Forets DML run on intervention ”social” on data from EU-SILC Norway, showing all deep generative models and the synthetic population resulting from generating regions by VAE-100 separately and then merge them into a population.



(a) Original

(b) VAE-15

(c) VAE-30

(d) VAE-50

(e) VAE-100

(f) WGAN-15

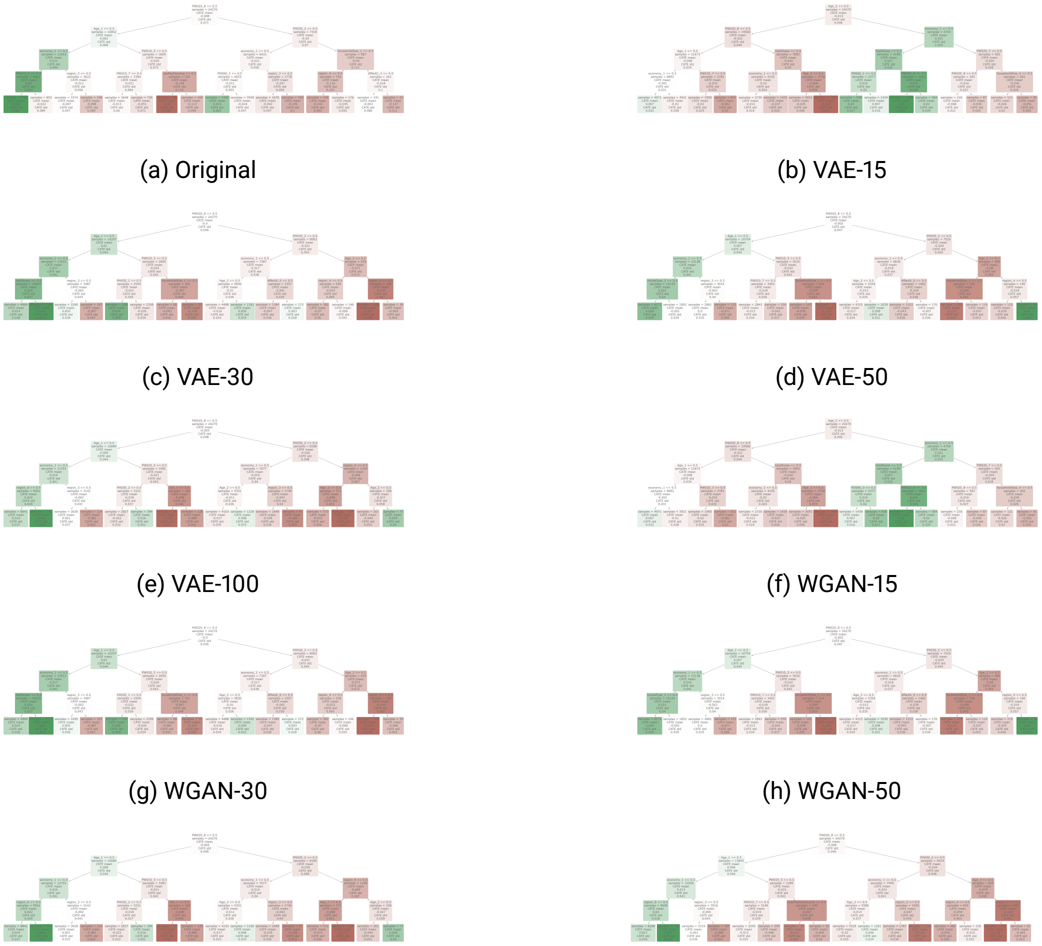
(g) WGAN-30

(h) WGAN-50

(i) WGAN-100

(j) Merged

Figure 79: Decision charts from Causal Forets DML run on intervention ”leisure” data from EU-SILC Norway, showing all deep generative models and the synthetic population resulting from generating regions by VAE-100 separately and then merge them into a population.



(a) Original

(b) VAE-15

(c) VAE-30

(d) VAE-50

(e) VAE-100

(f) WGAN-15

(g) WGAN-30

(h) WGAN-50

(i) WGAN-100

(j) Merged

Figure 80: Decision charts from Causal Forets DML run on intervention "affect" on data from EU-SILC Norway, showing all deep generative models and the synthetic population resulting from generating regions by VAE-100 separately and then merge them into a population.

NTNU

Norwegian University of
Science and Technology