Simulating Data: Generating Realistic Synthetic Data from Sampled Datasets

Karapet Ghazanjyan, Sedrak Yerznkyan

BS in Data Science, BS in Computer Science,

American University of Armenia

karapet_ghazanjyan@edu.aua.am

sedrak_yerznkyan@edu.aua.am

Supervisor – Karen Mkhitaryan

karen.mkhitaryan@clinstatdevice.com

***Abstract* —** This document gives brief understanding of the program that helps to generate synthetic data from the given input sample. The paper is going to discuss statistical methods used for the data generation, while trying to maintain statistical similarities to the original input data, which can be used for future algorithms testing and privacy protection of sensitive data.

Keywords— R, synthetic data, data generation, distribution fitting

I. Introduction

During the rise of artificial intelligence programs, machine learning algorithms, and with the rapidly growing data science field, the availability of large amounts of data has become crucial for practically any organization that wants to continue their growth. Data drives business decisions, improves customer experiences, and enables companies to remain competitive in the marketplace. However to be able to make such algorithms or decisions a huge amount of data is required to be analyzed, which sometimes may be a problem specially for small companies. The available data may not be sufficient, be limited, incomplete or contain sensitive information that cannot be used and shared for the purposes of a given project. In these and many other cases synthetic data can be a useful and cheap solution.

II. Methodology

The methodology of generating synthetic data with similar correlation and same distributions for each variable includes several steps. Using “fitdist” function from “fitdistplus” R package, we maximize likelihood estimate of each variable.

$$L(\theta) = \prod_{i=1}^{n} f(x_i|\theta)$$

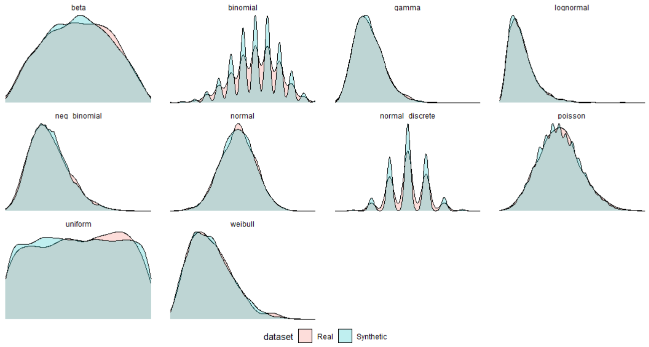
The The “fitdist” function returns numerical results, which also contain BIC, AIC and loglikelihood, which will be used to select the best fitting distribution for each variable. Depending on the method chosen for the distribution fitting the lowest (BIC, AIC) or the biggest (loglikelihood) values are being chosen from the “fitdist” function return to choose between given distributions.

After that correlation between all the variables are being calculated, which we use to create multivariate normal distribution. Using the correlation matrix obtained and “rmvnorm” function from “mvtnorm” R package, we create new dataset, which variables are normally distributed and correlated the same way as the original dataset. Next, using cumulative distribution function (CDF), the normally distributed variables are being transformed into uniform distributions. This process involves finding CDF of the multivariate normal distribution and applying values of the distribution to get uniformly distributed random variables.

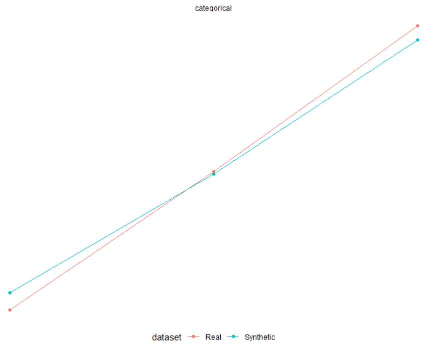
Finally by obtaining probability point function for each distribution, it should be applied to the uniformly distributed random variables generated in the above steps to obtain synthetic correlated variables having the same distributions and close to the original dataset distribution parameters.

III. Results and Discussion

As a way to test given methodology 2 dataset were used to understand the distributions of the variables and simulate new dataset with similar statistical parameters. First, dataset is created using R programming language and the distribution functions in base R. The dataset includes variables generated using binomial, negative binomial, poisson, beta, gamma, logarithmic normal, weibull, normal and uniform distributions. The purpose of this dataset is to be sure that the methodology helps to identify all the given datasets.

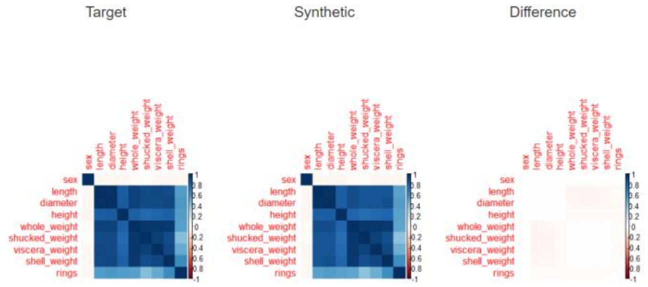


After running the dataset through the program and comparing density plots of input and output dataset, it is clear that it was able to clearly stimulate all the given variables and create variables with same distributions.

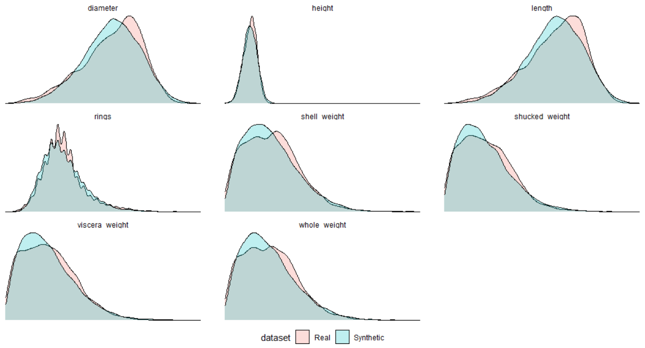


The same can also be concluded about the categorical variable, which also has proportions close to the original one.
But as there is no clear correlation between these randomly generated variables, another dataset from real world will be used to understand how the program behaves with dataset that has highly correlated variables.

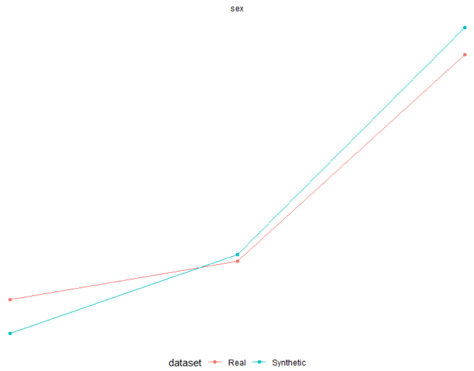
The second dataset is going to be taken from machine learning repository of University of California, Irvine.



The first correlation matrix shows the correlation of variables between input dataset, second one shows correlation matrix of synthetic data, which clearly mimics the original correlation matrix, and shows that given methodology works, the third matrix is the difference of two matrixes.



As with the first dataset, the density plots and proportions of categorical variables show that the program was able to clearly identify the best distributions for each variable and generate new ones for second dataset too, while also, keeping the correlations between the variables as close to the original one as possible.



IV. Conclusion

The process of generating synthetic data is an essential tool for many data science projects. This program demonstrated a methodology that can be used to generate synthetic data that preserves statistical properties of the original data. The data generated by this program can be used for sed for various purposes, such as testing statistical models, training machine learning algorithms, or ensuring data privacy. This study has brought attention to the significance of choosing appropriate probability distributions for fitting the data and assessing their attributes. The project also showed how well model selection criteria work in determining the appropriate distributions for each variable.

Overall, this study has offered a useful way for producing synthetic correlated data that may be utilized to protect data privacy while still enabling data analysis.

REFERENCES

1. Comprehensive R Archive Network (CRAN). (2023, April 25). Help to fit of a parametric distribution to non-censored or censored data [R package fitdistrplus version 1.1-11]. The Comprehensive R Archive Network. Retrieved May 5, 2023, from https://cran.r-project.org/web/packages/fitdistrplus/index.html
2. Multivariate Normal Distribution. Brilliant Math &amp; Science Wiki. (n.d.). Retrieved May 5, 2023, from https://brilliant.org/wiki/multivariate-normal-distribution/
3. Riemann: Rmvnorm – R documentation. Quantargo. (n.d.). Retrieved May 5, 2023, from https://www.quantargo.com/help/r/latest/packages/Riemann/0.1.1/rmvnorm
4. Sarkar, T. (2021, October 28). Synthetic Data generation -a must-have skill for new data scientists. Medium. Retrieved May 5, 2023, from https://towardsdatascience.com/synthetic-data-generation-a-must-have-skill-for-new-data-scientists-915896c0c1ae
5. The synthetic data generation and Knowledge Hub. MOSTLY AI. (2023, April 7). Retrieved May 5, 2023, from https://mostly.ai/
6. Synthetic Data: The Complete Guide. Datagen. (n.d.). Retrieved May 5, 2023, from https://datagen.tech/guides/synthetic-data/synthetic-data/
7. Tdistrplus: An R package for fitting distributions. (n.d.). Retrieved May 5, 2023, from https://mran.microsoft.com/snapshot/2015-08-14/web/packages/fitdistrplus/vignettes/paper2JSS.pdf
8. Turing. (2022, February 11). Synthetic data generation: Definition, types, techniques, &amp; tools. Synthetic Data Generation: Definition, Types, Techniques, &amp; Tools. Retrieved May 5, 2023, from https://www.turing.com/kb/synthetic-data-generation-techniques
9. UCI. (n.d.). Abalone Dataset. UCI Machine Learning Repository: Abalone Data Set. Retrieved May 5, 2023, from https://archive.ics.uci.edu/ml/datasets/abalone