AGILE: GIScience Series, 5, 26, 2024. https://doi.org/10.5194/agile-giss-5-26-2024
Proceedings of the 27th AGILE Conference on Geographic Information Science, 4–7 Sept. 2024.
Editors: Alison Heppenstall, Mingshu Wang, Urska Demsar, Rob Lemmens, and Jing Yao.
This contribution underwent peer review based on a full paper submission.

A software tool for generating synthetic spatial data for GIS-classroom usage

Paddy Gorry [1] and Peter Mooney [2]

[1]Hamilton Institute, Maynooth University, Maynooth, Ireland
[2]Department of Computer Science, Maynooth University, Maynooth, Ireland

Correspondence: Paddy Gorry (patrick.gorry.2015@mumail.ie)

Abstract. We describe a software tool for generating synthetic spatial data for use in GIS-related teaching and learning activities. Example deployments include the ability to provide every student with a different dataset(s) for a given spatial region or to quickly generate previously unseen datasets for a given spatial region. However, ensuring the datasets exhibit real-world characteristics is a much more difficult task. In this paper we report on ongoing work and our subsequent progress towards development of a reusable and reproducible software tool (written in Python) for the production of synthetic spatial data (vector polygons and points) for these purposes.

Keywords. Synthetic spatial data, randomised spatial data, software tool, teaching and assessment

1 Introduction

In GIS education, access to spatial data is essential for the learning and teaching process (Quinn, 2021). Suitable datasets may not always be available or may require significant time and technical resources to merge, integrate and prepare. Cobb (2020) argues that “these are technical but very real challenges within a field that is already highly labor-intensive”. To attempt to alleviate these problems we describe our current progress in the development of **a reproducible software-based tool for the generation of synthetic geospatial datasets (vector polygons and points) for use in GIS teaching and assessment**. While the specific stakeholder audience of our work contains both teachers and students on GIS-related courses we believe such synthetic spatial datasets could be used for spatial algorithm testing, development and testing of geospatial visualisation approaches, and so on. As a testbed for this development both authors are involved in the teaching of a long-running MSc module CS621B/C on Spatial Databases, at Maynooth University, Ireland, where the large class sizes necessitate a pedagogical approach embracing technology and automated approaches to teaching and assessment. Specifically, as outlined in section 2, we are developing this software to facilitate the generation of synthetic datasets for use by the students in both low-stakes learning tasks and high-stakes assessments. Crucially, for instructors, this tool will greatly reduce the amount of time and effort needed to create spatial data for classroom usage.

1.1 Contributions of this work

We define a synthetic spatial dataset as a collection of artificially generated geospatial objects, such as points, lines, or polygons, that are produced “from scratch” (Nikolenko, 2021) but informed by patterns of real-world data. The objects can optionally have attributes which in turn can be generated randomly or selected from predefined lists or rules. The software tool should be (a) easy to configure for non-specialists/programmers, (b) generate “realistic” synthetic data quickly for regions specified in valid GeoJSON polygons , (c) process data in standard geographic data file formats such as GeoJSON, ESRI Shapefile, and PostGIS SQL table dumps, and (d) require minimal or no access to externally available geographic datasets.

Our paper discusses the challenges and benefits posed by the usage and generation of synthetic spatial data for examinations and assessments in the GIS classroom setting. We position this within a specific teaching context (see Section 2). The paper extends our work on RADIAN (Gorry and Mooney, 2023) which describes a Python-based tool for synthetic spatial point data generation for GIS assessments. RADIAN provides educators with a means of producing customizable datasets for examinations and assessments that can be tailored to fit the needs and aims of the assessment. This paper describes the continued development of RADIAN to incorporate the generation of synthetic polygon spatial datasets. The generation of synthetic polygon datasets is significantly more difficult than the generation of synthetic points. RADIAN is already available as open-source software and the software code for this extended work is available as open-source software with the GitHub repository URL available at the end of this paper.

The remainder of our paper is structured as follows. In section 2 we provide some brief information on the teaching context inspiring this work. Section 3 provides some commentary on related work in this field. Section 4 and 5 describe approaches to generation of synthetic points and polygon datasets respectively. The paper closes with section 6 where conclusions and future work are discussed.

2 Teaching Context

Module CS621 "Spatial Databases" has been delivered as compulsory 7.5 ECTS credits module at our University since 2010. Module CS621 is compulsory on several different MSc. programmes within our University. Table 1 shows the number of students registered for Module CS621 since 2016. At the time of writing there were 79 students registered for the 2023/2024 academic year with the module being delivered in Semester 1 (Autumn semester) annually. In delivering this module, a great deal of instructor time and effort is put into developing and arranging assessments, both *for* learning, and *as* learning. The module runs over 12 weeks. For each weekly session there is at least one self-assessment (low-stakes and not counting towards final marks). Additional to these weekly self-assessments, there are four high-stakes lab exams during the semester. The weekly self-assessment exercises can be described as assessments *for* learning (Bennett, 2011), and can be useful for students to evaluate what their strengths and weaknesses in a module might be, while also allowing the teacher to understand what areas of the module may need more focus or re-working.

Table 1. Number of registered students on Module CS621 since 2016. October 2024 will be the next delivery of the module

Academic year	Number of students
2016	**30**
2017	**56**
2018	**60**
2019	**78**
2020	**81** (delivered online)
2021	**86**
2022	**66**
2023	**79**

Figure 1 illustrates the structure for the four high-stakes lab exams during the semester. An input spatial region R is specified within a GeoJSON file. In Figure 1 this region is a polygon around Nottingham, England. There are n students within the class. The software is then used to generate n synthetic datasets for this exam with each student $i \in n$ being provided with a different set of datasets for the input region R. Depending on the exam, each student i is provided with: a dataset of synthetic polygons P_i, a dataset of synthetic points p_i, or both P_i and p_i. The number of spatial objects within any dataset P_i or p_i is controlled by a set of global constants which set the minimum and maximum sizes of P_i or p_i. All students are provided with the same exam assessment sheet. In the case of module CS621 "Spatial Databases" solutions to exam assessment questions are usually SQL queries for PostGIS and/or visualisations/maps produced in QGIS. Solutions are submitted electronically. Exams are conducted in-person under normal timed and invigilated exam conditions. The exams are fully open book as this greatly reduces the stress on students during these assessments but also model the types of assessments found in many graduate training programs within industry (Cleophas et al., 2023; Green et al., 2016). Students can reference any of their material from the current module including model solutions, text, lecture materials, and so on. Use of Google, social media, ChatGPT, StackOverflow, and so on are not allowed.

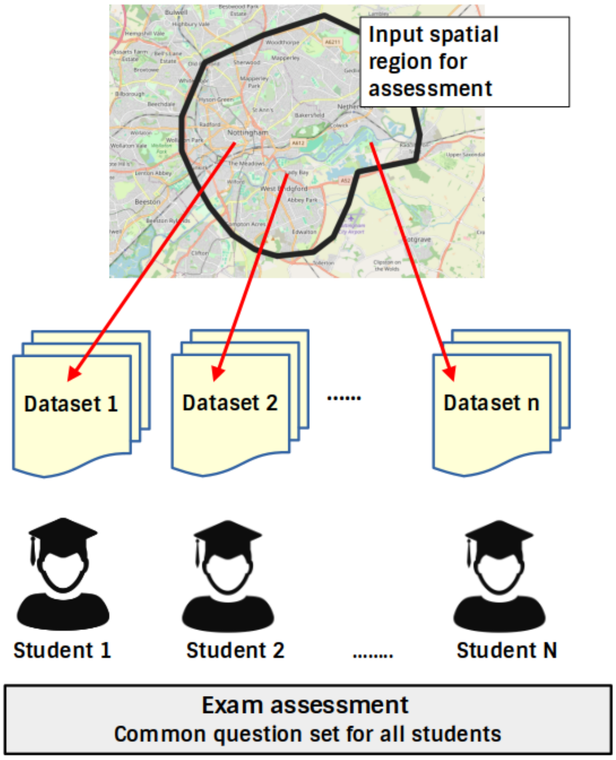


Figure 1. Exam assessment interaction diagram

3 Review of related work

The ability to generate synthetic spatial data quickly and easily can greatly reduce the time and effort required by both teachers and students to access suitable data for teaching and learning tasks. Preparing spatial datasets for classroom usage often takes considerable time and effort. Hagge (2023) argues that teaching GIS and related topics have many costs associated with hardware and software requirements. However, he points out that there are many "nonmonetary costs". These costs are incurred from the the technical expertise required as "teaching GIS at any level of instruction can be more time-consuming than other academic disciplines". Consequently, teachers of GIS and related topics often have to develop assignments on their own. This work can take away precious in-class time that would otherwise be spent teaching geography content (Tan and Chen, 2015). Previous work, such as that by Welle Donker et al. (2022), has shown that the flexibility for teachers to use or generate datasets that will be relevant and relatable to students help to improve overall student engagement with the learning materials.

Much of the literature relating to synthetic data generation focuses on machine-learning approaches to the subject. Generative adversarial networks (GANs) allow for the learning of a data distribution through competitive training between two neural networks. Cunningham et al. (2022) demonstrated using a GAN for producing synthetic spatial data with the aim of maintaining data subject anonymity through the use of local label differential privacy. Xiao et al. (2017) also implemented a GAN which aimed to learn spatial point process models. Quick et al. (2015) also worked with point processes for geographic data but using Bayesian framework. In similar work Mannino and Abouzied (2019) demonstrates a tool for producing "real-looking" synthetic data allowing users to specify the statistical distribution, among other parameters, of their datasets. However the focus of this tool is on multivariate numerical data and does not consider spatial data. There appears to be little work reported on the generation of synthetic polygons. Zhu et al. (2022) outlines an algorithm for generating building polygons based on an image inputs. However, this is different to the concept around our software tool as there is a need for significant external data to be available for approaches like this.

4 Points: Generating Synthetic Points

Points generation can be considered as two problems: (1) generating a set of points and their coordinates and then (2) ensuring those coordinates are within in the desired input region R. The latter issue refers to the famous "point-in-polygon" computational geometry problem (Hormann and Agathos, 2001) and relates to how one decides if a given point p lies within the bounds of the specified polygon R. For our context, it is likely that the region R would refer to an arbitrary geographical region or administrative boundaries.

4.1 Generating Synthetic Points - Approach

RADIAN uses the following approach for generating synthetic points datasets. Given a source region R and a set of parameters (such as maximum $Nmax$ and minimum $Nmin$ number of points, attributes for each point object) the generation of points takes place at two scales: at the level of the whole region R, and at the level of i smaller sub-regions $r_i \in R$ generated within R. Point coordinates are sampled from a uniform distribution within the boundary of R. A series of Voronoi-polygon buffers are generated around the centroid of R. Overlapping buffers of increasing area are used to produce a more centralized distribution. Each buffer b is assigned an equal number of points b_n. This generates a distribution that concentrates towards the centroid of R with the density decreasing towards the outer boundary of R. One of the main computational challenges in this process is generating points within R quickly and efficiently. Computationally, point-in-polygon operations can be very expensive for large R and large $Nmax$. In our software we use spatial joins provided by the `Shapely` and `GeoPandas` to significantly reduce processing times.

This approach, used in RADIAN, produces synthetic spatial datasets that present a spatial distribution informed by examples seen in real-world data. Many metropolitan areas will have geographic features distributed around a central area or region whereby point features are concentrated (Yang and Hillier, 2007; Peponis et al., 1997), with their density reducing as distance to this central area increases. On a surface level, this creates realistic datasets, however on closer inspection this assumption begins to fail. We are currently working to improve this aspect of the work. At large geographic scales, such as R representing the Greater London area for example, this approach generates more heavily clustered maps particularly when $Nmax$ is large. RADIAN generates points based on a uniform distribution, where coordinates are generated randomly and removed if they do not lie within R. Presently, RADIAN does not, however, take into account real-world context, such as the locations of roads, rivers, or land-use categories. A benefit of this approach is the need for extetnal datasets is eliminated, however we can encounter some basic problems. Our current work is also addressing this issue by adding in the ability to understand the semantics of the positioning of a given randomly generated point. An example Fig 2 displays as set of 1000 points that, at this scale, appear to be distributed in an appropriate, realistic manner. When the scale changes however, as shown in Fig 3, we can see that the lack of extra semantic contextual information results in points landing in rivers and roads, which in many applications would be unrealistic and unacceptable. However, in our teaching context (as outlined in Section 2, this is not a particularly impactful issue where students are learning to write distance queries, kernel density estimation calculations, choropleth mapping, and other GIS topics. RADIAN can use `.csv` files with a list of possible attribute values, and optional weights, that can then be used to assign real-world attributes (such as restaurant names) to the generated points. An example is shown in Fig 4

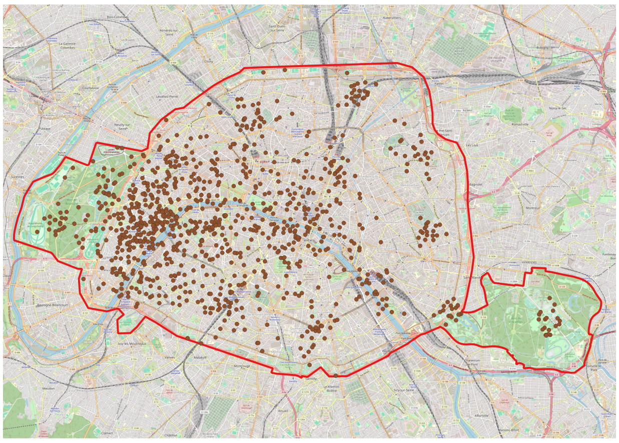


Figure 2. Sample dataset of 1,000 points generated in Paris

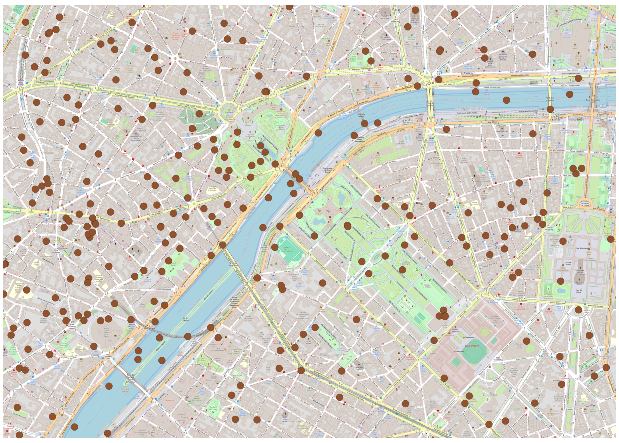
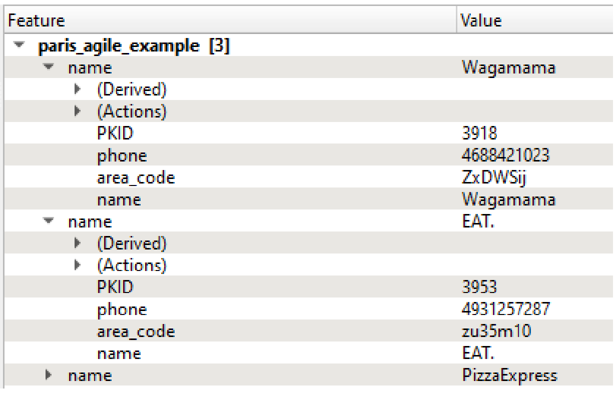


Figure 3. Paris, showing the lack of context in generation



Feature	Value
▾ paris_agile_example [3]	
▾ name	Wagamama
▸ (Derived)	
▸ (Actions)	
PKID	3918
phone	4688421023
area_code	ZxDWSij
name	Wagamama
▾ name	EAT.
▸ (Derived)	
▸ (Actions)	
PKID	3953
phone	4931257287
area_code	zu35m10
name	EAT.
▸ name	PizzaExpress

Figure 4. Example metadata generated using RADIAN

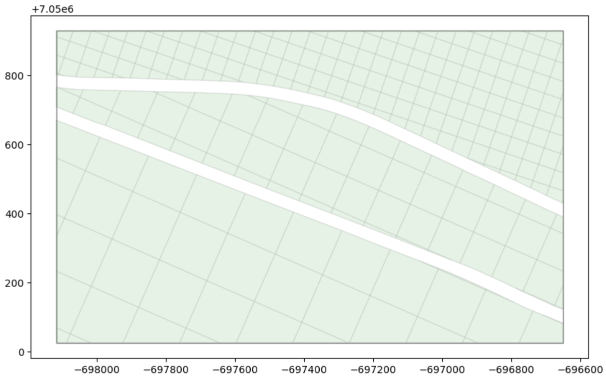


Figure 5. Rotated tilings with railway buffers

5 Polygons: Generating Synthetic Polygons

The task of generating point objects, as described in Section 4, is relatively straightforward but not without its complexities. Polygons, on the other hand, are made up of multiple point objects (vertices) joined by linestrings (edges) and are more complex to generate randomly. Greater consideration must be given to their placement in relation to other polygons in the same dataset. As before, our task is to take a polygon region R and then automatically generate up to m polygons within R. We previously experimented with Voroni diagrams for the purpose of generating the m polygons within R however, this approach required us to generate a point dataset as part of this process. We wanted to develop an approach which did not require the generation of additional data for the synthetic polygons.

5.1 Generating Synthetic Polygons: Tiling and Buffering

Tiling of a polygon refers to the partitioning of a polygon into sub-polygons. With `Shapely` and `GeoPandas` it is relatively straightforward to produce a tiling of a given polygon using repeated regular shapes such as squares or triangles. Viewing the tile maps as an overlay on a real-world map, this lack of rotation leads to strange and confusing visualisations. Through trial and error we found that it was necessary to utilise some external source of data to generate synthetic polygons that looked “realisitic”. To achieve this, we automatically download polygon data from OSM (in the region R), specifically way data, which is used to describe the paths of roads/rail/waterways. The example shown in Fig 5 shows how tilings of varying size and rotation can be generated in the sub-regions within R to provide a more varied distribution.

Given a desired tile square width and angle of rotation, we generate an amount of squares that will completely cover the given region R. These tiles can then be clipped, using Shapely, to remove any squares or portions of squares that are not within the region R, as shown in Fig 5. The OpenStreet Map road data is then loaded in LineString format, and the Shapely `.buffer()` command is used to create buffer regions, B, along the length of the roads. Finally the GeoPandas `.overlay() function` is called to subtract the intersection between B and R. The result of this overlay, as shown in Fig 6 shows a set of points generated within these polygons. This displays how these buffer regions can be used to restrict the area of a given polygon region, in this case it ensures that point objects generated in this area will not be able to lie directly on roadways.

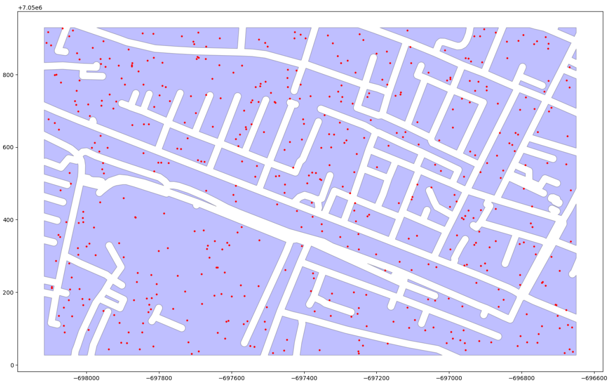


Figure 6. Valid regions created with road buffering

5.2 Polygons: Generating Synthetic Building Polygons

Partitioning, tiling, or creating a Voroni diagram of a given polygon region R is reasonably straightforward with available tools (see section 5.1). However, anecdotally, after some experimentation, we found that such approaches can generate unrealistic sets of polygons. This becomes more problematic as the number of sub-polygons required within R grows large. Subsequently, in this stage of the software tool development, we decided to focus on the generation of synthetic polygons representing building footprints in urban areas. To produce realistic synthetic building polygons geographic and semantic context are required and it becomes impossible to avoid access to external data for this purpose. We use the OpenStreetMap (OSM) API in order to download all *real* data (lines, polygons, and points) within the specified region R. By keeping R small this is download is completed quickly.

To generate the synthetic building polygons we make the assumption that the synthetic buildings will also adhere to the configuration of the real road network in the region R. The set of polygons generated in Fig 6 are reused. The Shapely `.buffer()` function is applied to each individual polygon, as in Step 1 in Fig 7. A vector grid, as described in section 5.1, is then generated over the newly generated buffers. The GeoPandas `.overlay()` function is applied again. this time acquiring the intersection between the newly generated buffers and vector grid, as shown in Step 2 of Fig 7. The remaining polygons in this overlay are then subject to another buffering operation. This step ensure that there is a separation between each building polygon, as in Step 3 of Fig 7. Finally, in Step 4 of Fig 7, we sample a set of n polygons (in this case we set $n = 600$) from the newly generated synthetic building polygons. To avoid selection of very small polygons we add a configurable constraint to avoid selection of polygons with an overall area of $< 75m^2$. The metadata generated by RADIAN demonstrated in Section 5 can be applied in the same way to the generated building polygons, allowing for richer resulting datasets as the buildings can have specific attributes, names, or addresses assigned to them.

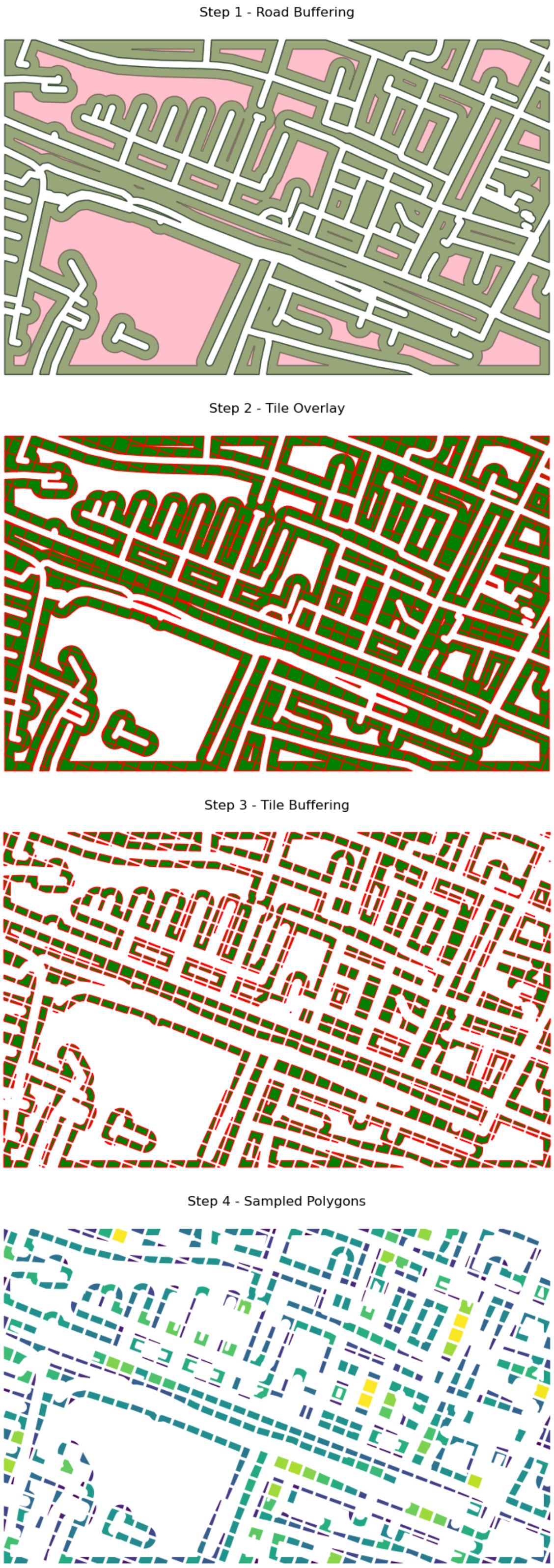


Figure 7. Creation of synthetic building polygons through use of polygon tiling, OSM roadway data, and buffering

6 Conclusions and Future Work

This paper has described our current work in the development of a reproducible software-based tool for the generation of synthetic geospatial datasets (both vector polygons and points). Currently, this tool is being tested and used within a classroom setting for an MSc-level module on Spatial Databases. While the generation of synthetic datasets of points (see section 4) is progressing well, the generation of synthetic datasets of polygons (see section 5) is a more complex problem and is currently only at the early stages of development. However, we believe our initial examples in section 5 are promising but require further work. There are a number of significant contributions from this work, including:

- Overall, this tool shall present a very useful pedagogical tool for both teachers and students alike by greatly simplifying the preparation of spatial data for learning as well as assessment and examinations.
- The approach to point and polygon generation are both scalable to different geographical regions and different numbers of objects within those regions.
- The tool allows additional attribute metadata to be accessed and assigned to point and polygon objects allowing these synthetic datasets to engage students with real-world examples.
- There is very significant reduction in the overall time and resources required to prepare spatial datasets for GIS-related modules. The Python-based tool can run without any external datasets.
- The software can be used independently by students both in their own study time and within exam time to generate spatial datasets (as shown in Figure 1)

Our discussions in this paper describe our current progress towards the delivery of a software tool which can be more widely used for the creation of useful, realistic, and reproducible spatial datasets within learning and learning contexts. There are several issues under investigation as future work. Work by Quick and Waller (2018) uses a fully Bayesian hierarchical model to directly model data with exact geographic locations and both categorical and non-categorical attributes. Using an approach such as this, certainly for point data, would allow the generation of synthetic data by fitting a statistical model in a Bayesian setting. Considering more deeply spatial autocorrelation (Klemmer et al., 2019), for example, would potentially introduce more realistic behaviours in the spatial distributions of points generated by this tool. There is an increasing focus on ML and AI methods for performing some of these tasks (as discussed in section 3) we are attempting to use an algorithmic approach which we believe will make the tool easier to set up, run, and configure for a wide range of stakeholders including teachers and students. The potential to develop a QGIS-plugin will also be investigated when the software tool has reached a more stable point in its development.

Reproducible research statement

The work shown in this paper is freely available and reproducible through the RADIAN Git repository - https://github.com/paddeaux/radian.

References

Bennett, R. E.: Formative assessment: a critical review, Assessment in Education: Principles, Policy & Practice, 18, 5–25, https://doi.org/10.1080/0969594X.2010.513678, 2011.

Cleophas, C., Hönnige, C., Meisel, F., and Meyer, P.: Who's cheating? mining patterns of collusion from text and events in online exams, INFORMS Transactions on Education, 23, 84–94, 2023.

Cobb, C. D.: Geospatial Analysis: A New Window Into Educational Equity, Access, and Opportunity, Review of Research in Education, 44, 97–129, https://doi.org/10.3102/0091732X20907362, 2020.

Cunningham, T., Klemmer, K., Wen, H., and Ferhatosmanoglu, H.: GeoPointGAN: Synthetic Spatial Data with Local Label Differential Privacy, 2022.

Gorry, P. and Mooney, P.: A reproducible approach to generating synthetic spatial data for teaching and learning purposes, in: Proceedings of the 31st GISRUK Conference, pp. 123–136, https://doi.org/10.5281/zenodo.7825100, 2023.

Green, S. G., Ferrante, C. J., and Heppard, K. A.: Using Open-Book Exams to Enhance Student Learning, Performance, and Motivation., Journal of Effective Teaching, 16, 19–35, 2016.

Hagge, P. D.: GIS in the Non-GIS Classroom: Using Student Mapping Assignments to Incorporate GIS in Traditional Lecture Classes, The Geography Teacher, 20, 50–56, https://doi.org/10.1080/19338341.2023.2233521, 2023.

Hormann, K. and Agathos, A.: The point in polygon problem for arbitrary polygons, Computational geometry, 20, 131–144, 2001.

Klemmer, K., Koshiyama, A., and Flennerhag, S.: Augmenting correlation structures in spatial data using deep generative models, 2019.

Mannino, M. and Abouzied, A.: Is this Real? Generating Synthetic Data that Looks Real, in: Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology, UIST '19, p. 549–561, Association for Computing Machinery, New York, NY, USA, https://doi.org/10.1145/3332165.3347866, 2019.

Nikolenko, S. I.: Synthetic data for deep learning, vol. 174, Springer, 2021.

Peponis, J., Ross, C., and Rashid, M.: The structure of urban space, movement and co-presence: The case of Atlanta, Geoforum, 28, 341–358, 1997.

Quick, H. and Waller, L. A.: Using spatiotemporal models to generate synthetic data for public use, Spatial and Spatio-temporal Epidemiology, 27, 37–45, https://doi.org/https://doi.org/10.1016/j.sste.2018.08.004, 2018.

Quick, H., Holan, S. H., Wikle, C. K., and Reiter, J. P.: Bayesian marked point process modeling for generating fully synthetic public use data with point-referenced geography, Spatial Statistics, 14, 439–451, https://doi.org/https://doi.org/10.1016/j.spasta.2015.07.008, 2015.

Quinn, S.: Using free and open source software to teach university GIS courses online: Lessons learned during a pandemic, The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 46, 127–131, 2021.

Tan, G. C. I. and Chen, Q. F. J.: An Assessment of the Use of GIS in Teaching, Geospatial technologies and geography education in a changing world: Geospatial practices and lessons learned, pp. 155–167, 2015.

Welle Donker, F., van Loenen, B., Keßler, C., Küppers, N., Panek, M., Mansourian, A., Zhou, P., Vancauwenberghe, G., Tomić, H., and Kević, K.: Showcase of Active Learning and Teaching Practices in Spatial Data Infrastructure (SDI) Education, AGILE: GIScience Series, 3, 1–11, 2022.

Xiao, S., Farajtabar, M., Ye, X., Yan, J., Song, L., and Zha, H.: Wasserstein learning of deep generative point process models, Advances in neural information processing systems, 30, 2017.

Yang, T. and Hillier, B.: The fuzzy boundary: the spatial definition of urban areas, in: Proceedings, 6th International Space Syntax Symposium, İstanbul, 2007, pp. 091–01, Istanbul Technical University, 2007.

Zhu, Y., Huang, B., Gao, J., Huang, E., and Chen, H.: Adaptive Polygon Generation Algorithm for Automatic Building Extraction, IEEE Transactions on Geoscience and Remote Sensing, 60, 1–14, https://doi.org/10.1109/TGRS.2021.3081582, 2022.