TOWARDS AN OPEN SOURCE PYTHON LIBRARY FOR AUTOMATED EXPLORATORY SPATIAL DATA ANALYSIS

Nicholas de Kock[1,*], Victoria Rautenbach[1], Inger Fabris-Rotelli[2]

[1] Department of Geography, Geoinformatics and Meteorology, University of Pretoria, Pretoria, South Africa – nicholas.dekock@tuks.co.za, victoria.rautenbach@up.ac.za
[2] Department of Statistics, University of Pretoria, Pretoria, South Africa – inger.fabris-rotelli@up.ac.za

Commission IV, WG IV/3

KEY WORDS: Open Source, Python library, Spatial Statistics, ESDA

ABSTRACT:

The exploratory spatial data analysis (ESDA) process refers to the use of various functions to gain an initial understanding of a spatial dataset. These include measures of spatial heterogeneity and spatial autocorrelation. Currently, the ESDA process is repetitive and time-consuming. Additionally, while different results arise for different datasets, how these results are generated does not change significantly. Results are also generated individually for each variable which means that they cannot be easily compared or shared.
The automation of the ESDA process would therefore have multiple benefits as it would not only save time, but it would also allow the data analyst to keep up with the rapid rate at which we generate data. This paper aims to introduce the first iteration of *autoESDA* – a Python library capable of automating the ESDA process by summarising the results into a single report.
In this paper, we present the defined high-level requirements for the implementation of *autoESDA*. Various dependency libraries are discussed and a high-level overview of the workflow of *autoESDA* is described. The library is then evaluated against the requirements laid out earlier in the study. Semi-structured interviews were carried out, which yielded a wealth of feedback and suggestions from the participants, describing how the output report could be improved. Finally, a roadmap of proposed further developments and improvements is discussed.
The first version demonstrates that the automation of ESDA is possible and lays the foundation for further development in this regard. This is an important contribution to understanding spatial data as it enables the data analyst to keep up with the magnitude of data that is generated on a daily basis.

1. INTRODUCTION

In recent years there has been a largescale increase in the volume of spatial data generated. This exponential increase in both the volume and velocity of spatial data is attributed partly to the decreasing price of sensors, along with a world where topics such as the "Internet of Things" and "big data analysis" have experienced dramatically increased popularity (Armstrong, Wang, and Zhang, 2019). Spatial data is rapidly created through methods such as the geotagging of images on social media and traffic data from users of navigation software such as Google Maps. While there is great benefit in the availability of datasets, true value can only be obtained once this data is processed into useful information.
The data lifecycle refers to numerous stages that result in the transition of raw data into information. The lifespan of a data lifecycle varies according to the dataset (Raju and Nathan, 2018). Exploratory data analysis (EDA) is a process carried out near the beginning of a data lifecycle. Its purpose is to gain a basic understanding of the dataset. For spatial datasets, this process is known as exploratory spatial data analysis (ESDA) (Dall'erba, 2009).
ESDA is made up of various functions that aid the exploration of spatial datasets and identification of patterns that may otherwise go unnoticed (Murray and Estivill-Castro, 1998). Results arising from the ESDA process often dictate how the data is further utilised. ESDA functions can be carried out on both vector and raster based spatial data (Moura and Fonseca, 2020). The current iteration of *autoESDA* only supports data in vector polygon format, however work is currently underway to extend this functionality to support raster and other vector formats.
Two important components of the ESDA process are spatial autocorrelation, and spatial heterogeneity. Spatial heterogeneity is investigated using choropleth maps, box plots, scatterplots, and histograms, for example, allow for one to identify trends or patterns, that could have otherwise gone unnoticed. The results of spatial autocorrelation are vital, as they dictate whether or not the recorded instances of a phenomenon are spatially dependant on each other (Dall'erba, 2009).
The ESDA process is both repetitive and time-consuming and the automation thereof would allow the data analyst more time to be able to focus on more important aspects of the data lifecycle. In practice, ESDA functions are run individually, meaning that results are also displayed individually, which does not allow for comparisons to be made.
There are numerous open source technologies with ESDA capabilities; the three most popular ones are Python libraries, R libraries, and GeoDa. Python has numerous advantages when working with spatial data, such as its ability to handle large datasets. Various libraries also allow Python to easily integrate with most geoportals, spatial database management systems, and other GIS technologies (Cura, 2019). Python libraries such as *pandas*, *geopandas*, *pysal*, *plotly* and *matplotlib* are often used for executing ESDA functions and displaying these results. The seamless integration and wealth of available libraries make Python an ideal choice for automating the ESDA process.
Automation of similar processes using Python are not unheard of. The EDA process has been automated using Python libraries such

* Corresponding author

as *pandas-profiling*, *sweetviz*, and *autovis*. These libraries allow a data analyst to easily carry out the EDA process by executing one line of code, which generates an HTML report that neatly summarises the results.

Automation of various processes within the spatial data lifecycle are not uncommon. The curation of spatial metadata falls under the transformation stage of the lifecycle (Ciceli, 2015); it has been automated by Batcheller (2008). The generation of thematic maps falls under the distribution stage of the spatial data lifecycle and various efforts have been made to automate this process (Coetzee and Rautenbach, 2017; Pillay et al., 2019). While these examples are not entirely related to ESDA, they do, however, illustrate that there is a benefit to automating repetitive processes within the spatial data lifecycle.

The aim of this paper is to present our first iteration of *autoESDA*, a library that automates the ESDA process in Python. The paper discusses the design and implementation of the library by describing the high-level requirements, dependency libraries, and workflow of the library itself. The library is then evaluated according to the defined requirements, and numerous interviews are conducted in order to get feedback on the first iteration. Finally, a roadmap for further development is discussed.

2. REQUIREMENTS AND IMPLEMENTATION

2.1 Requirements

Multiple high-level functional and non-functional requirements were defined during the planning phase of this project. These requirements were decided on by identifying solutions to the major issues encountered by carrying out an ESDA process and what a potential solution would look like. Once the library was developed, the specified requirements were revisited to ensure that the high-priority requirements were satisfied. As this was an iterative process, if the functional requirements were not satisfied, the researcher returned to the development phase to ensure that all the high-priority requirements were met.

Functional requirements refer to functions that a system is required to perform (Young, 2003). Conversely, non-functional requirements refer to properties of a system that do not dictate, what needs to be done, but rather how well it should be done. Table 1 summarises the high-level functional and non-functional requirements that were defined for the development of the *autoESDA* library.

Functional Requirement	Description
Report Output (High Priority)	The library should generate an HTML report that can be saved to the local computer.
Spatial Heterogeneity (High Priority)	The generated report should include a boxplot, histogram, descriptive statistics, and correlation statistics.
Spatial Autocorrelation (High Priority)	The generated report should include a Moran's I simulation, the associated statistics, and a LISA cluster map.
Data Type Detection (High Priority)	The library should be able to distinguish between columns that can be plotted and have statistics calculated on them. It is assumed that data is already converted into the correct types and unsupported data types (such as strings and characters) should be ignored.
Non-Functional Requirement	**Description**
Simple Execution (High Priority)	The library should be simple to use, meaning that only one parameter (the GeoDataFrame) is required to generate the report.
Offline Availability (Low Priority)	The generated report should not reference any external sources, this would mean that the report does not require an internet connection to display correctly.
Colour Use (Medium Priority)	A suitable colour scheme/theme for the report should be selected that is both appealing and free from any alternate connotations.
About Page (Medium Priority)	The report should include an about page which tells the user what defaults have been set for the generated figures and statistics.
Data Sample (High Priority)	A subset of the dataset should be displayed in the report.
Performance (Low Priority)	The library should generate a report timeously once the function has been executed.

Table 1. High Level Requirements

2.2 Design

The first aspect of the design stages focused on deciding which ESDA functions to include in the library. These decisions were made according to how popular a certain ESDA function was, as well as how easily it could be automated. Functions that require the data analyst to specify numerous parameters that could not easily be set to a default setting, are assumed to be more difficult to automate.

Most ESDA functions are simple enough to automate; however, parallel coordinate plots (PCPs), measures of autocorrelation, and choropleth maps are all seen to be complex to automate. This is due to the fact that they have numerous parameters that need to be specified. The combination of variables to include in the PCP will depend greatly on the dataset used and a generic solution for automation cannot easily be implemented (Zhou et al., 2018). The same argument can be made for the automation of spatial autocorrelation and choropleth maps, as the input parameters could have a huge effect on the outputted results.

The functionality to allow the user to specify their own spatial weights matrix has not been included in the current version of *autoESDA*. Spatial autocorrelation, however, is seen as a high priority requirement, meaning that there needs to be some degree of spatial autocorrelation in the output report, and the utility thereof can be evaluated in the interviews. For this version of *autoESDA*, it was decided that a Moran's I simulation with a queens first-order matrix along with a LISA map would be included in the report as measures of global and local spatial

autocorrelation. Moran's I was the chosen function as it is the most commonly used measure of spatial autocorrelation (Jackson et al., 2010). A first order queens case matrix was used as the default spatial weights matrix.

Due to their importance in visualising the spatial distribution of different variables, the decision was made to include choropleth maps. To overcome the need to specify which classification scheme to use for these maps, the decision was made to include four choropleth maps, each with a different classification scheme, for each variable. It was also decided to use the *geopandas* default number of intervals, which is five.

With the exception of those mentioned above, the majority of ESDA functions do not require input parameters and could therefore be automated with relative ease. These include generic five number summaries (minimum, mean, median, maximum, and standard deviation), boxplots, histograms, scatterplots, and correlation matrices.

There are numerous Python libraries that exist with the intention of solving various problems or addressing different needs within the Python development community. In the development of this library, existing functions from other libraries were used.

Table 2 describes the libraries referenced in *autoESDA*, which are known as dependencies. Each of these libraries have been included as they serve a specific purpose in the *autoESDA* library. These libraries have been chosen according to the functions which they provide, and their relative popularity. Choosing libraries according to popularity has two main advantages, namely: readily available support, and a community of contributors who help to ensure updates and bug fixes are routinely rolled out.

Dependency (Version)	Description
geopandas **(0.8.1)**	The *geopandas* library is an extension of the popular *pandas* library which defines data frames as a way to structure data. *geopandas* adapts this as a way to store spatial data such that this GeoDataFrame is the attribute table, where there are additional columns for geometry or coordinates.
libpysal **(4.4.0)**	This is the core library that *pysal* is based on. It is used in this project to create the spatial weights matrix which is used in the autocorrelation calculations.
pysal **(2.3.0)**	This library, along with its dependencies, allow for the plotting of the choropleth maps as well as the Moran scatterplot and LISA cluster map.
matplotlib **(3.4.2)**	*matplotlib* is a popular library for creating graphs and other visual aids. This library enables the use of grids and annotations to combine the numerous figures together.
seaborn **(0.11.2)**	*seaborn* is similar to matplotlib, however it has extra functions such as the heatmap and pairplot() function which was used in this project.
io **(3.8.10)**	This library converts images into objects made up of bytes. In conjunction with the *base64* library, it allows for images to be embedded/stored directly in the HTML file.
base64 **(3.8.10)**	The *base64* library was used to encode images as objects and works in conjunction with the *io* library to enable storage and embedding of images within the HTML file. This avoids the need for external files.

Table 2. Dependencies of the autoESDA library

2.3 Implementation

Once the decisions were made regarding which ESDA functions and dependencies to use, it was time to design how the functions would work together to generate an appropriate report. This workflow is visualised in Figure 1.

To begin with, the library will accept a GeoDataFrame, from which it will determine which columns have a numeric data type and which do not. The ESDA functions for *autoESDA* are calculated from numeric data, which is why this differentiation needs to take place.

Both non-numeric and numeric data is required for the Summary Page, which can be seen in Figure 2. This is because the Summary Page displays a sample of the dataset, as well as a description of which datasets were included in the report (numeric datasets) and which datasets were not included (non-numeric datasets). In addition to the sample of the dataset, the Summary Page also includes a basic outline of the study area, descriptive statistics, and some basic metadata such as projection used.

The next block of code entails a loop which iterates in order to create a variable information page for each numeric column in the GeoDataFrame. Each iteration of the loop will create a boxplot, histogram, various choropleth maps and a Moran's I and LISA simulation. An example Variable Information Page is shown in Figure 3.

Finally, the Correlation Page (shown in Figure 4), composed of a heatmap and pairwise plot, was created using the numeric variables. The Correlation Page, along with the summary page and all the variable information pages was combined into an HTML report, which was then saved to the working file directory.

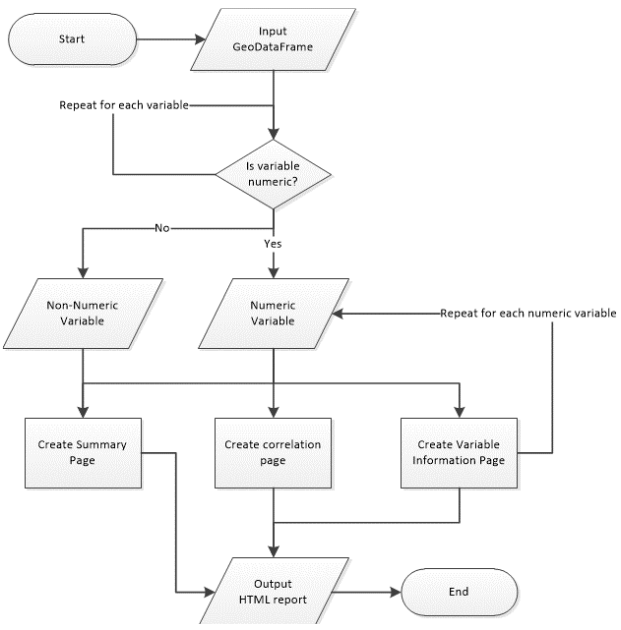

Figure 1. Workflow of the *autoESDA* library

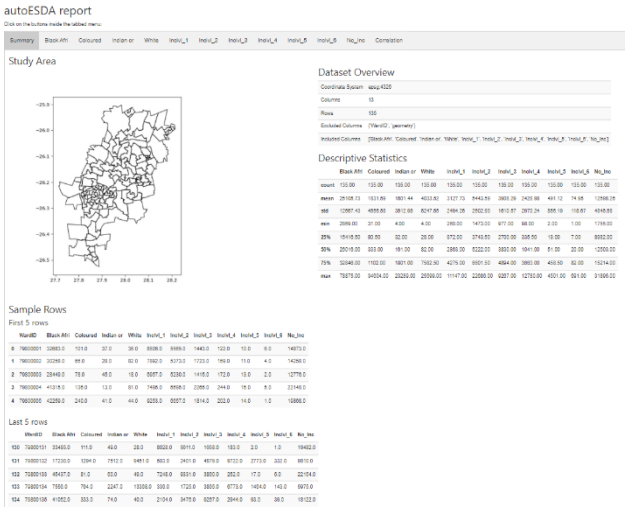

Figure 2. Summary Page

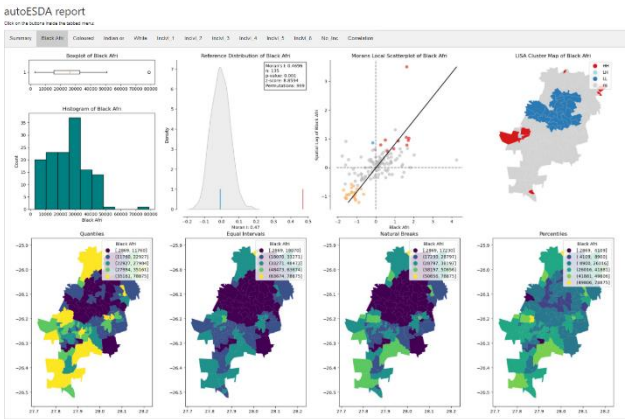

Figure 3. Variable Information Page

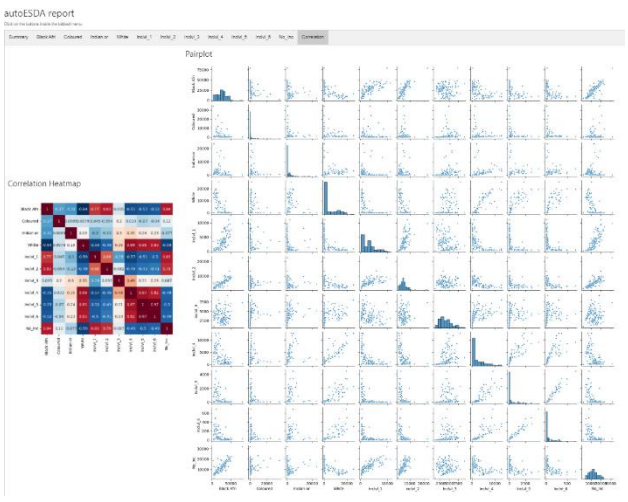

Figure 4. Correlation Page

2.4 Availability and Usage

The source code for *autoESDA* can be found at: https://github.com/autoESDA/autoESDA-static. This code is available under the BSD 3-Clause license. An example report generated by *autoESDA* can also be found at: https://autoesda.github.io/autoESDA-static/.

3. EVALUATION

3.1 Evaluation against requirements

In Section 2.1, four functional requirements and six non-functional requirements were defined. This was used as a guide for the researcher to gauge how much development still needed to take place for the first iteration of the library. This section discusses how each of the requirements were met, whereas the interviews and discussion investigate how well these requirements were met.

3.1.1 Functional Requirements

The library **generates an HTML report** which is saved to the working directory, thus satisfying the first functional requirement.

Measures of **spatial heterogeneity** include a descriptive statistics table which has a count, mean, standard deviation, minimum, 25th percentile, median, 75th percentile, as well as a maximum value for each variable. Furthermore, there are boxplots, histograms, and numerous choropleth maps for each variable, as well as a correlation matrix.

Spatial autocorrelation has been addressed through the inclusion of a reference distribution (to evaluate the statistical significance of the calculated values), Moran's scatterplot, and LISA cluster map. The reference distribution plot also displays the Moran's I value, sample size, p-value, z-score, and number of permutations.

The final functional requirement refers to the library's ability to **discern numerical variables** from the rest. This is because mathematical plots and statistics can only be generated from data that is numeric in nature. This is done in the first few lines of code of the library.

3.1.2 Non-functional Requirements

The first non-functional requirement which has a high priority is that the library runs using only **one line of code**, making it simple to use. While the library can be called using one line of code, it is currently not a published library which means that this requirement is not entirely satisfied. The library currently accepts no parameters except for the GeoDataFrame itself.

Offline functionality was a low priority non-functional requirement, which was not satisfied through the development of this library. The requirement aims to ensure that the report can be viewed without an internet connection, however this is not the case as it references two external style sheets. While the report will still display when offline, the experience for the user may be different.

The use of **appropriate colour schemes** that are neither conflicting, nor misleading, were listed as a medium priority non-functional requirement. This is a challenging requirement to meet as it could be very subjective in nature. The researcher made all possible efforts to choose suitable colour schemes and the chosen colours were put through a colourblind simulator. The colours used deemed to not be misleading or conflicting and are appropriate for people with colour vision impairment. For this reason, it is argued that the requirement for suitable colours has been met, and the extent to which this requirement has been satisfied will be determined through the interview process.

Other non-functional requirements include the presence of an **About page**, describing decisions and default values made in order to generate the report, as well as a **sample of the original dataset**. The current iteration of autoESDA does not have an About page, however there are plans to include this in future

iterations. The library does, however, show the first and last five rows as a sample of the dataset.

Performance is the final non-functional requirement. It was listed as low priority and were not tested. In order to test performance, a benchmark needed to be identified – this benchmark has not yet been decided on. As such, it cannot be said if or how well this requirement was satisfied. There are, however, plans to test the performance of *autoESDA* in the future.

3.2 Interview process

Numerous interview participants with varying experience, careers, and frequency of using ESDA were used in the interview process in order to generate a variety of feedback. Table 3 summarises the demographic information of these participants. The interviews that took place were semi-structured in format. Table 4 shows the predefined questions which were used as a guide for the interview process. This was seen as the most effective strategy to adopt as it allowed the researcher to gain a further understanding regarding some statements that were made by the participants.

Participant	Years of experience	Sex	Industry	Job title	How often do you use ESDA functions?
1	2	M	Software Engineering	Software engineer	Never
2	20	M	GIS education	Associate professor	Monthly
3	4	F	Commercial GIS	Geospatial consultant	Monthly
4	1	M	Commercial GIS	Data Scientist	Monthly
5	6	M	Commercial GIS	Geospatial Developer	Every Two months
6	1	M	GIS	Geoinformation specialist	Never
7	2	F	GIS	Student Assistant	Monthly
8	24	M	GIS/Cartography	Senior Cartographer	Weekly
9	25	M	Education	Freelance data analyst	Monthly
10	8	M	GIS & research	GIS analyst, Lecturer	Weekly
11	4	M	IT/ Data science	Data scientist	Monthly
12	17	F	Research	Associate professor	Never
13	1	F	Research	Lecturer	Never

Table 3. Demographics of the interview participants

The interviews were carried out on the Zoom video conferencing platform as this allowed for the researcher to share their screen and eliminated the need for any travel or physical meetings between the researcher and the participants. This also allowed for a wider variety of participants as travel was not necessary. The participants were sent an example report beforehand, so that they had time to look at it and consider some feedback before they were interviewed.

Interview questions
1. What position do you hold, and how does it require you to make use of ESDA functions?
2. What challenges do you currently have when conducting an ESDA process?
3. Could you tell us about the process you follow when you are performing ESDA process?
[Show prototype]
4. General
a. How comfortable are you using Python?
b. Can you think of any improvements that could be made to the structure/layout of the report?
c. Are the titles of each section clear or are they misleading i.e., do you get the information you expect when selecting them?
d. For each page, could you say whether this section useful to you? What improvements would you recommend?
e. Now that you have seen what the library can do, would you use it where necessary?
f. What hesitations do you have about the use of this library?
5. Summary page
a. Are there any other features/statistics you would like or expect to be on the summary page?
b. Are the statistics on the summary page useful to you?
6. Choropleth maps
a. Are there extra classification schemes for choropleth maps that you would like to be included in this library?
b. Would you prefer there to be more/less classes for the choropleth maps?
c. Do you feel that the colour scheme is suitable? If not, do you have a recommendation as to what it should be?
d. Would you recommend any other improvements to be made to the choropleth maps section?
7. Autocorrelation
a. Are there any extra statistics you would expect to find in a report like this?
b. Do you feel that it is necessary to include the probability distribution and scatterplot?

	c. A queens contiguity matrix with an order of one has been set as the default, do you feel that this is a good idea? Is there another strategy which you would prefer?
	d. Are there other important autocorrelation measures that you would prefer to Moran's I?
8.	Correlation
	a. How easy is it for you to interpret the correlation matrix/heatmap?
	b. Do you think it is necessary to include the scatterplots for each relationship?
	c. Do you think the colour scheme is suitable? If not, do you have a recommendation as to what colour scheme should be used?
9.	Pairwise plot
	a. Do you like the layout of the pairwise plot or do you find it confusing to understand?
	b. What colour scheme do you feel should be used for the pairwise plot?
	c. How many bars do you think a histogram should have?

Table 4. Interview questions

3.3 Interview Feedback

There were thirteen participants who gave feedback, each with different academic backgrounds, work experience and experience levels. The variety of the backgrounds of each participant lead to a huge amount of varied and sometimes contradictory feedback.

This feedback is divided into four sections, namely: the Summary Page, the Variable Information Page, the Correlation Page, and the About Page. All of the participants were impressed with the library prototype and concur that the progress has been in the right direction. Participant 2 said that the report was "*great, and very useful*" which was supported by Participant 3 who said that the tool filled a "definite need in the GIS industry". Participant 5 stated that the report was "*useful and well implemented*" which backs up the opinion of Participant 10 in that "*everything I looked for was here*".

3.3.1 Summary Page

The first major element on the Summary Page (as shown in Figure 2) is the **map of the study area**. In general, the participants were glad that it was present and provided the user with some insight about the shape of the area that is described in the report. Participants 2-6 all indicated that they see value in this map being interactive, with popups providing them with the relevant information for each of the polygons when hovered over. Participant 7 also recommended the use of colour in the study area map in order to make it more appealing. While this would improve the library's appearance, the issue would be selecting an appropriate colour scheme that does not have any potential connotations depending on the datasets used in the report. It was also suggested that there should be a name of the study area above the map. This may be challenging due to the versatility of the library being able to generate generic reports, however it was suggested that the call function of the library should have a parameter where the user could specify a name.

Participant 2 who comes from a GIS education background, mentioned that students may be confused by the use of the terms rows and columns as it is too similar to raster data, and that the terms attributes and fields should be used instead.

There was not much feedback given from the participants relating to the **dataset overview table** with the exception of participants 5 and 11 who mentioned that they would like to see some spatial statistics included in it. Examples they gave included average area of the polygons and average number of neighbours.

The other major element on the Summary Page was the **descriptive statistics table**. In general, the participants were satisfied that most statistics were present with the exception of the skewness, kurtosis, as well as the number of null or unique values in each column. These majority of participants made this comment. In addition to this, Participant 2 also suggested that the descriptive statistics table include a Moran's I value.

The final element on the Summary Page is the **dataset sample** which consisted of the first and last ten rows to give the user an idea of what the original dataset looked like. There were numerous contrasting views amongst the participants regarding what constitutes a suitable sample. Participants 1, 9, and 11 were of the opinion that showing 20 rows was excessive and that only the first and last five rows were necessary. Participant 2, however felt that all the rows in the dataset should be included and that the user should be allowed to query these in order to aid their understanding of the dataset. Strategies such as only showing the first ten rows, or a random ten rows were also suggested by Participants 4 and 5.

3.3.2 Variable Information Page

The first two elements on the Variable Information Page (as shown in Figure 3) were the **boxplot** and the **histogram**. No major comments were received from any of the participants; however, each of them emphasised the importance of having these present. Once asked about the number of bins recommended for the histogram, the participants seemed to be happy with the default value of ten bins and did not see the need for this to change.

The **reference distribution** drew quite a lot of feedback from the participants. While it is a good inclusion in the report, the lack of a key for the red and blue lines on the diagram, coupled with the non-descriptive title, gave some of the participants the impression that it could be improved. The x-axis and y-axis could be more descriptive such as explaining. The values in the textbox could also be coloured red or blue to link them to the line on the reference distribution that they relate to. One of the participants also suggested that a "clustered/not clustered" label should be included on the reference distribution. Some of these changes would be challenging to implement as it would involve the modification of code in the existing *pysal* library.

One the major issues identified with the **Moran's scatterplot** was that the visual gradient of the line of best fit does not match the Moran's I value (this should not be the case). This has been brought about by the stretching of the scatterplot to match the size of the other subplots; however, this is misleading towards the user. One of the participants also commented on the colours used in the scatterplot, citing the fact that the user is not told what these colours represent, and is unsure as to whether they relate to the LISA scatterplot or not.

The participants also indicated that they valued the inclusion of the **LISA cluster map** as part of the report. There was, however, a comment on the colour scheme chosen with one of the participants having the opinion that a single, graduated colour scheme would be more suitable than the Red-Blue colour scheme currently being used. Some of the participants also found the labels in the legend to be difficult to understand, and that inexperienced users may not understand that *HH* and *ns* represents a polygon that has a High-High neighbourhood classification or a relationship that is not significant with its neighbours.

All of the participants were of the opinion that **Moran's I** was an appropriate measure to be used as a measure of autocorrelation, rather than another measure such as Geary's c. While the participants did not express strong opinions regarding what spatial weights format was the most appropriate, they indicated that the default of a queens contiguity with an order of one was acceptable, provided that this was indicated somewhere. Participants 9, 10 and 11 all indicated that they would like the functionality that would allow them to specify their own spatial weights matrix as a parameter of the call function for the report.

The **choropleth maps** generated a lot of discussion, with majority of the feedback being directed towards the legend placement, which covered a large portion of the map. Although the matplotlib parameter of best-position is used, it is evident that the placement is not always optimal. Some suggestions to overcome this from the participants included placing the legends outside the map, removing the decimals (which are unnecessary) from the legend and making it a horizontal rather than a vertical legend. It was also mentioned that the variable name should be included in the title of the map and not in the legend. Regarding the classification schemes chosen, the participants were in general happy with those that were present, however some participants did suggest a box map and standard deviation classification scheme to also be included. When questioned about the number of classes for maps (currently the default of 5 is used), none of the participants considered this to be a problem. The colour scheme was also mentioned in the interviews, with the majority of the participants happy with the current one being used. One comment regarding the colour scheme which arose from two of the participants was that it should be inverted, so that values with a greater magnitude are assigned the darker, more intense colours.

Participant 10 questioned if the report was suitable for those who are colourblind. This was not a consideration in the lifecycle of the project, and it was decided to test the report using Colblinder, an online colourblind simulator. Red and green colour-blindness are the most common types, and this is what was simulated. The results show that there is an effect of these types of colour-blindness, however all features still vary enough to the colourblind eye to be differentiated from each other.

Concluding remarks relating to the Variable Information Page were that it feels very congested, and that this could be avoided by increasing the spacing between plots, and by removing the borders from the choropleth maps.

3.3.3 Correlation Page

The Correlation Page (as show in Figure 4) was made up of a correlation heatmap and a pairplot. The majority of the participants found benefit in there being both a correlation heatmap as well as a pairwise plot. Participants 3, 6, 9, and 8 who are all very experiences in the GIS industry, suggested that the values in the **correlation heatmap** should be rounded up to two decimal values. The colour scheme of the heatmap was also discussed, with some participants of the opinion that it is too closely related to the colours used in the autocorrelation subplots, and therefore misleading. Participants 12 and 13 stated that a colour ramp outside of the correlation matrix would improve their understanding. Importantly, Participants 10-13 also questioned which type of correlation was used as it was not stated anywhere, and that a user should be able to choose which correlation measure they would like to be present in the report.

While captioned as a **pairplot**, it was brought to the attention of the researcher that the diagram should more appropriately be called a pairwise plot. The pairwise plot could be made more user friendly through the use of more labels, and red borders for the subplots with significant relationships (correlation values above |0.7|). Statistics such as coefficient of determination, trendlines and adjusted R^2 values would also be of value to the user. One of the participants also stated that a correlation value is not suitable if the data is not linear, and for this reason it may be beneficial to include a warning for relationships that are non-linear yet are found to have a significant correlation.

Finally, the placement and layout of the Correlation Page drew a reasonable amount of discussion. Some of the participants preferred both the pairwise plot and the heatmap to be square in shape and rather placed under each other for more space. Other participants, however, were of the opinion that only the upper or lower triangle were necessary and that instead these two elements should be combined so as to maximise the use of space, while minimising the duplication of information on the page. This would be a valuable improvement, however due to the pairwise plot being a function from the *seaborne* library, these suggestions would be difficult to implement.

3.3.4 About Page

While this page was not in the prototype shown to the participants, there was a lot of discussion around the necessity of an About Page and, therefore, it has been given its own subheading. The About Page should act as a manual for the generated report that users could navigate to so that they may improve their understanding of the report. Some elements that were suggested to be included here were the number of histogram bins, significance values and spatial weights used for the Moran's I simulation, description of each of the subplots, number of default classes for the choropleth maps and correlation type used in the correlation heatmap. Additionally, Participant 9 suggested that the data and time that the report was generated be included, as well as a disclaimer relating to how the report should be used.

4. ROADMAP OF FURTHER DEVELOPMENTS

The development of *autoESDA* is an ongoing process, meaning that the current version has laid the foundation for more features to be included in the future.

A major improvement to *autoESDA* will be the ability to accept multiple data formats. Currently, the library only works with vector polygon geometries, however there is scope for this to be improved to support vector line and point data, as well as data in raster format.

Results from the interviews are discussed in Section 3, and highlight multiple opportunities for further developments. An example of this is the ability of the user to specify their own spatial weights matrix. This means that instead of using the current default of a queens first-order matrix, the used could specify as a parameter the shape and order of their preferred spatial weights matrix.

Interview participants also mentioned that including additional ESDA functions such as Geary's c, would add to the wealth of information in the generated report.

Participants had numerous suggestions that would improve the layout of the report. These suggestions included repositioning some of the elements, as well as increasing the spacing between figures so that the report does not feel so congested.

A popular suggestion amongst most interview participants was the inclusion of an About Page in the report. The About Page would provide the user with information relating to the *autoESDA* library, as well as the report metadata such as the date generated, the default values for choropleth maps, or the type of spatial weights matrix used in the autocorrelation simulation.

Testing the scalability and performance of the library is another aspect of *autoESDA* that is planned for future work. This includes investigating how efficient the script is in processing datasets, as well as if it has the capability to handle large volumes of data with the same efficiency.

One important milestone planned for the *autoESDA* library is the refactoring of the code so that it may be used in a QGIS plugin. This will eliminate the need for a used to have a knowledge of Python and will allow the user to generate an *autoESDA* report through a graphical user interface on the popular GIS platform.

5. CONCLUSION

The aim of this research was to present a first iteration of *autoESDA*. This was achieved by describing the process of defining requirements and designing the library's workflow. *autoESDA* was then evaluated against the predefined requirements, as well as through the use of interviews to generate feedback. While the first iteration of *autoESDA* is functional, there is planned improvements and additional functionality. Aspects of the library such as scalability and performance could also be investigated to ensure that the library is capable of handling the large datasets that are common in today's data-driven world. This article presented the first iteration of *autoESDA* and in doing so, has laid the foundation for more work to be carried out in the automation of the ESDA process.

REFERENCES

Armstrong, M., Wang, S., Zhang, Z., 2019. The Internet of Things and fast data streams: prospects for geospatial data science in emerging information ecosystems. Cartography and Geographic Information Science 46, 39–56. https://doi.org/10.1080/15230406.2018.1503973

Batcheller, J., 2008. Automating geospatial metadata generation—An integrated data management and documentation approach. Computers & Geosciences 34, 387–398. https://doi.org/10.1016/j.cageo.2007.04.001

Coetzee, S., Rautenbach, V., 2017. A Design Pattern Approach to Cartography with Big Geospatial Data. Cartographic Journal 54, 301–312. https://doi.org/10.1080/00087041.2017.1400199

Cura, R., 2019. Enriching Exploratory Spatial Data Analysis with modern computer tools, in: European Colloquium of Theoretical and Quantitative Geography - ECTQG 2019. Mondorf-les-Bains, Luxembourg.

Dall'erba, S., 2009. Exploratory Spatial Data Analysis, in: Kitchin, R., Thrift, N. (Eds.), International Encyclopedia of Human Geography. Elsevier, Oxford, pp. 683–690. https://doi.org/10.1016/B978-008044910-4.00433-8

Moura, A.C.M., Fonseca, B.M., 2020. ESDA (Exploratory Spatial Data Analysis) of Vegetation Cover in Urban Areas—Recognition of Vulnerabilities for the Management of Resources in Urban Green Infrastructure. Sustainability 12, 1933. https://doi.org/10.3390/su12051933

Murray, A., Estivill-Castro, V., 1998. Cluster discovery techniques for exploratory spatial data analysis. International Journal of Geographical Information Science 12, 431–443. https://doi.org/10.1080/136588198241734

Pillay, L., Schaab, G., Coetzee, S., Rautenbach, V., 2019. A comprehensive workflow for automating thematic map geovisualization from univariate big geospatial point data. International Cartographic Association 2, 8. https://doi.org/10.5194/ica-proc-2-100-2019

Raju, P., Nathan, B., 2018. The Data Life Cycle. Strategic Finance 100, 62–63.

Young, R., 2003. Requirements Engineering Handbook. Artech House, Norwood, United States.

Zhou, Z., Ye, Z., Yu, J., Chen, W., 2018. Cluster-aware arrangement of the parallel coordinate plots. Journal of Visual Languages & Computing 46, 43–52. https://doi.org/10.1016/j.jvlc.2017.10.003